



CAMPUS
DE EXCELENCIA
INTERNACIONAL



Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Escuela Técnico Superior
de Ingenieros Informáticos

TRABAJO FIN DE GRADO

Marco de diseño de la interacción
para aplicaciones móviles

Autor: Abimael Barea Puyana

Director: Xavier Ferré Grau

MADRID, ENERO DE 2015

Resumen

El desarrollo de aplicaciones móviles multiplataforma se ha convertido en una tarea compleja debido al alto crecimiento de dispositivos móviles y a la diversidad de plataformas que existen en el mercado. Tras identificar que las principales plataformas utilizadas por los usuarios a nivel mundial son iOS y Android y observar la evolución que han sufrido dichas plataformas desde un punto de vista de diseño de la interacción en los últimos años se establecen las bases para la realización del marco de diseño de la interacción en aplicaciones móviles.

El marco de diseño pretende identificar la estructura de las aplicaciones en ambas plataformas y observar las similitudes y diferencias que existen entre las mismas. Por otra parte, se analizan, como parte del mismo, los elementos presentes en el diseño de la interacción de aplicaciones en las plataformas estudiadas con el fin de establecer similitudes y equivalencias entre los mismos.

Para evaluar la viabilidad del marco propuesto, se ha llevado a cabo un caso de estudio en el que se aplica el marco sobre una aplicación real. Por último, se realiza una herramienta web con la que se permite a los usuarios poder acceder a la información propuesta en el marco de una forma más sencilla.

Abstract

The mobile app development field has become a very complex task due to the increase of mobile devices and the diversity of the market existing platforms. After identifying that the most used platforms by the users are iOS and Android, and observing the evolution of both from the point of view of the interaction design, the basis for an interaction design frame are established.

The design presents the basic structure of a mobile application in both platforms, identifying the existing similarities and differences among them. Moreover, the elements present in the design of the interaction are analysed, in order to establish similarities and equivalences among those elements.

In addition, a case study was carried out in which the design frame was applied in a real application. Finally, a web tool which allow users to access the proposed information was developed, in order to have an easier access to the information.

Contenido

1. Introducción	1
1.1. Estructura del documento.....	1
2. Diseño de la interacción iOS vs Android.....	3
2.1. iOS.....	3
2.1.1. Evolución en el diseño de aplicaciones.....	3
2.1.2. Filosofía de diseño	7
2.1.3. Elementos para el diseño de la interacción en iOS.....	8
2.2. Android.....	9
2.2.1. Evolución en el diseño de aplicaciones.....	9
2.2.2. Diversidad de versiones y capas de personalización	13
2.2.3. Filosofía de diseño	16
2.2.4. Elementos para el diseño de la interacción en Android.....	18
3. Desafíos del diseño multiplataforma.....	20
3.1. Punto de vista técnico.....	20
3.2. Punto de vista del diseño de la interacción	21
4. Marco de diseño de la interacción en aplicaciones móviles: Navegación y distribución de elementos	23
4.1. Niveles de navegación	23
4.2. Distribución de elementos dentro de la ventana.....	27
5. Marco de diseño de la interacción en aplicaciones móviles: Criterios de clasificación	30
5.1. Comparativa de elementos	31
5.2. Análisis de elementos	32
5.2.1. Navigation	32
5.2.2. Bars	34
5.2.3. Content Views.....	35
5.2.4. Controls	36
5.2.5. Progress & Activity.....	39
5.2.6. Temporary Views.....	40
6. Caso de estudio	44
6.1. Estructura de la aplicación	44
6.2. Análisis de elementos	47
7. Herramienta web.....	49
7.1. Tecnologías web	49
7.1.1. Responsive Design	49
7.1.2. Análisis de frameworks para Responsive Design	50
7.2. Diseño de la herramienta	51
7.2.1. Estructura.....	51
7.2.2. Caso de estudio	52
8. Conclusiones.....	55
9. Líneas futuras	56
10. Referencias	57

1. Introducción

En los últimos años la utilización de dispositivos móviles, tipo smartphone, ha sufrido una creciente evolución haciendo que los usuarios cada vez demanden más este tipo de productos. Las aplicaciones que se utilizan en estos dispositivos han ido adaptándose a los avances tecnológicos y a las capacidades hardware que ofrecen dichos dispositivos. Debido a la gran demanda existente, en el mercado existe una alta diversidad de plataformas software, entre las cuáles encontramos: iOS, Android, Windows Phone, BlackBerry y otras con menor número de usuarios. Todo esto lleva a que para que una aplicación llegue al mayor número de usuarios posibles sea necesario que la misma esté presente en múltiples plataformas.

En el presente trabajo se abordan las complejidades que conlleva el desarrollo de una aplicación multiplataforma, centrándonos en las plataformas de aplicaciones móviles iOS y Android, ya que ambas abarcan a la amplia mayoría de usuarios de smartphones. Este tipo de desarrollo plantea desafíos en el diseño de la interacción, tanto debido a las distintas convenciones y guías de diseño de cada plataforma, como en cuanto a los elementos disponibles para utilizar en la interfaz de usuario.

Por todo esto, el objetivo principal del presente trabajo consiste en la elaboración de un marco de diseño de la interacción para aplicaciones móviles que facilite la toma de decisiones de diseño considerando los objetivos específicos de cada proyecto y teniendo siempre presente garantizar la mejor experiencia a los usuarios, intentando para ello que las aplicaciones sean acordes al resto dentro de la plataforma a la que pertenecen. Para conseguirlo, el marco aborda factores que van desde la estructura de las aplicaciones hasta los elementos usados en las diferentes partes de las mismas.

El marco desarrollado se ha probado sobre una aplicación móvil real, como caso de estudio, donde se analizará la validez del mismo para la evaluación de aplicaciones móviles existentes y su aplicación en proyectos de desarrollo móviles reales. Para la realización de dicho caso de estudio se ha utilizado como referencia la aplicación de la Escuela, que se encuentra en fase de desarrollo.

Por último, se ha desarrollado una herramienta web que permite visualizar la información contenida en el marco de una forma sencilla, para facilitar de este modo la difusión del mismo.

1.1. Estructura del documento

El presente documento sigue la estructura descrita a continuación:

- El presente capítulo es introductorio, y da una visión global de lo que se trata a lo largo del documento.

- En el segundo capítulo se analiza cual es el estado actual del diseño de la interacción en aplicaciones móviles para las plataformas iOS y Android.
- El tercer capítulo aborda los principales desafíos que conlleva el desarrollo de aplicaciones multiplataforma.
- En el cuarto capítulo comienza el desarrollo del marco de aplicaciones, donde se analiza cual es la estructura de una aplicación móvil.
- El quinto capítulo continua con el estudio del marco de aplicaciones, pero esta vez analizando las similitudes y equivalencias de los elementos presentes en las diferentes plataformas.
- En el sexto capítulo se evalúa la aplicación móvil de la Escuela mediante el uso del marco de diseño de la interacción en aplicaciones móviles.
- En el séptimo capítulo se detallan las tecnologías web utilizadas para el desarrollo de la herramienta y las características de la misma.
- En el octavo y noveno capítulo se presentan las conclusiones del presente trabajo y se plantean cuales deberían ser las líneas futuras sobre las que habría que trabajar para la continuación y ampliación del presente trabajo.

2. Diseño de la interacción iOS vs Android

Para comprender cuál es la situación actual del diseño de la interacción en aplicaciones móviles y la filosofía de diseño de cada una de las plataformas es necesario conocer la evolución que han experimentado en los últimos años las plataformas iOS y Android. Por ello, en el presente capítulo, se realizará un repaso sobre la evolución del diseño de la interacción en ambas plataformas desde sus orígenes hasta la actualidad.

2.1. iOS

El sistema operativo iOS ha sufrido grandes transformaciones y una gran evolución desde su aparición en el año 2007. Concretamente, y desde el punto de vista de diseño de la interacción, la evolución ha venido marcada por la introducción de nuevos elementos o tamaños de pantalla, pero al cambiar de la versión 6 a la 7 se llevó a cabo el mayor cambio hasta el momento ya que se modificó la filosofía de diseño y la estética de las aplicaciones.

2.1.1. Evolución en el diseño de aplicaciones

Durante las primeras 6 versiones del sistema operativo iOS la estética de las aplicaciones fue la misma y venía marcada por el *skeuomorph*.

El *skeuomorph* es una filosofía de diseño que fue promovida desde la primera versión del sistema por Steve Jobs, CEO de la Apple en aquel momento, y Scott Forstall, vicepresidente senior de software iOS en aquel momento. Dicha filosofía consistía, según [McNeil, 2013], en definir objetos con elementos de decoración que provienen del original. Esta técnica es usada para referenciar al original y por tanto crear un objeto que resulte familiar.

A partir de iOS7 en el año 2013, tras el cambio de CEO y al poner a Jony Ive, jefe de diseño, encargado de la interfaz de iOS se produjo un cambio en la filosofía, dejando atrás el *skeuomorph* y buscando el diseño de interfaces más minimalistas. En una entrevista Jony Ive dijo: “tenemos una gran libertad no referenciando al mundo físico de forma tan literal, estamos intentando crear un entorno que sea menos específico” [Cava, 2013].

Para entender mejor en qué consistía el *skeuomorph* vamos a ver unos ejemplos de aplicaciones iOS que seguían esta filosofía. La aplicación de calculadora en las versiones anteriores a la 7 es un claro ejemplo de cómo las aplicaciones intentaban emular elementos existentes en la realidad ya que estaba claramente inspirada en la calculadora ET66 de la firma alemana Braun diseñada por Dieter Rams y Dietrich Lubs en el año 1987 [Bloq, 2013].



Figura 1. Inspiración para calculadora

Otra aplicación que tiene una clara inspiración en la realidad es el Game Center que intenta emular las mesas de juego tradicionales de los casinos como podemos observar a continuación.



Figura 2. Inspiración para Game Center

En el año 2013, tras la presentación de iOS7, se publicó una Guía de Transición de Interfaz de Usuario de iOS7 desarrollada por Apple [Apple, 2014b], donde se explica de

forma detallada cuál es la evolución que han sufrido las interfaces. A continuación se va a hacer mención a algunas de las principales características a tener en cuenta.

En la Figura 3 se puede observar la evolución en el diseño sobre la aplicaciones de notas entre la versión iOS6 e iOS7. Se pasó de intentar emular al papel y la escritura a mano sobre un cuaderno a tener un diseño minimalista, con un único color de fondo, donde los iconos y elementos del sistema destacan en otro color. Estos elementos han sido simplificados, utilizando la geometría y finas líneas. La tipografía que se usa también ha sido renovada, denominándose Helvetica Neue, y sigue el mismo estilo simple que el resto de los elementos del sistema.

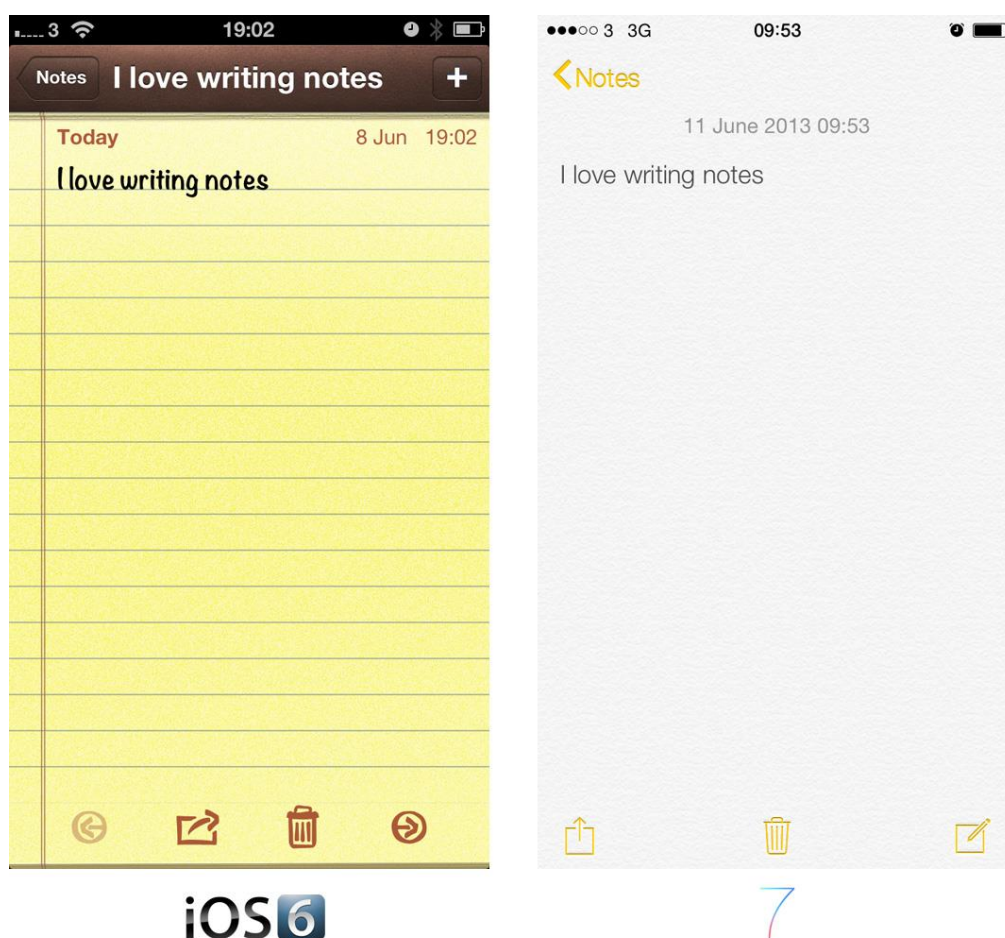


Figura 3. Comparación app notas

Otro ejemplo más concreto sobre la evolución que han sufrido los elementos del sistema son las barras, en este caso podemos ver cómo los márgenes se han simplificado y las sombras de los elementos que ayudaban a crear una impresión de relieve y volumen se han sustituido por elementos planos. Además, se han eliminado los degradados del fondo sustituyéndolos por transparencias y efectos como el *blur*, que consisten en una

especie de desenfoque de las imágenes o elementos que forman parte del fondo de la ventana.

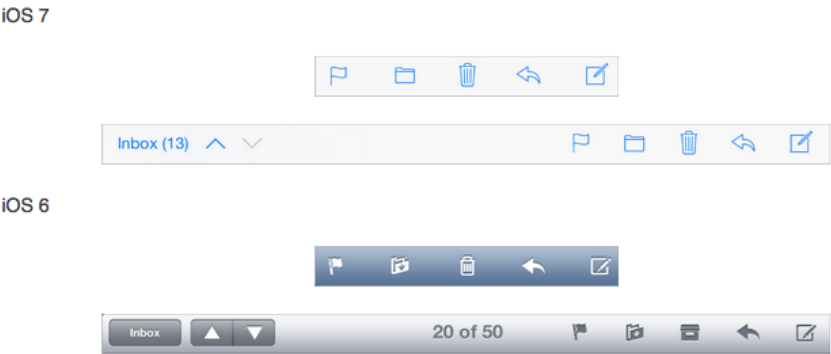


Figura 4. Comparación de barras

Las tablas también han sufrido una notable evolución, eliminando la textura del fondo por un color plano y uniforme, además de sustituir los bordes redondeados por ángulos rectos más acordes a la nueva filosofía en la que los elementos siguen formas geométricas.

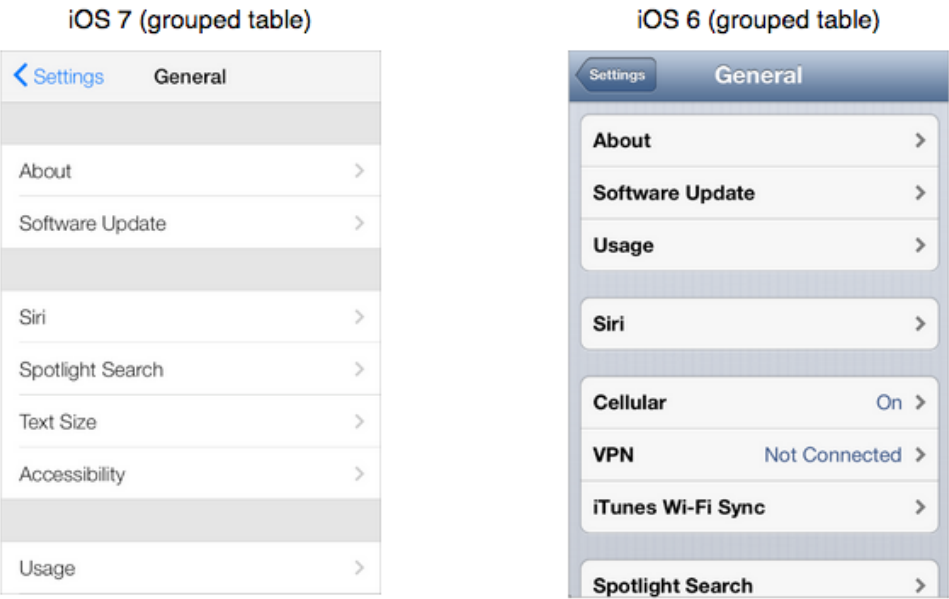


Figura 5. Comparación de tablas

2.1.2. Filosofía de diseño

Como punto de partida de la filosofía de diseño, es necesario entender cómo se realiza la navegación entre las distintas ventanas que componen una aplicación. En esta plataforma, la navegación entre los distintos subniveles de una aplicación, debe ser clara y sencilla otorgando al usuario una referencia de en qué ventana se encuentra en el momento actual y de dónde viene. Esto se consigue mediante el uso de una Navigation Bar situada en la parte superior de la pantalla que contiene, un texto indicativo de la ventana actual, en la parte central, y de forma ocasional un botón, en la parte de la izquierda de la barra, que sirve para navegar hacia la anterior ventana, es decir, ir hacia atrás. Para volver a la pantalla de inicio del sistema operativo o cambiar de aplicación se utiliza un botón físico que incorpora el dispositivo, por lo que el uso de la Navigation Bar está restringido a la navegación dentro de una aplicación, como se puede observar en la Figura 6.

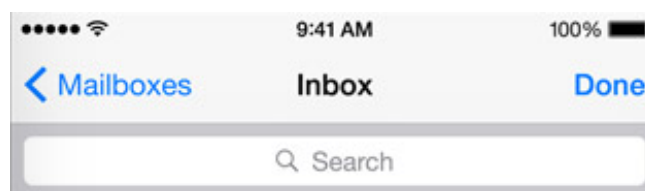


Figura 6. Barra de Navegación iOS

Adicionalmente y tras estudiar la guía de diseño de Apple [Apple, 2014a], se han considerado transcendentales algunos puntos clave en la filosofía de diseño de la plataforma, recogidos de diferentes secciones de la guía de diseño. Estos son los siguientes:

- Razón: la interfaz ayuda al usuario a entender e interactuar con el contenido, pero compite con el.
- Claridad: el texto debe ser legible, los iconos deben ser precisos, los adornos apropiados y el enfoque debe estar en la funcionalidad.
- Profundidad: las capas visuales y movimientos realistas imparten vitalidad y aumentan placer y comprensión por parte de la gente.
- Contrastes: ten en cuenta los contrastes de color en los diferentes contextos de la aplicación.
- Elementos estándar: utiliza los elementos estándar del sistema de forma correcta para que tu aplicación se integre con el resto del ecosistema iOS.

2.1.3. Elementos para el diseño de la interacción en iOS

Cada una de las plataformas tiene sus propios elementos que sirven para la composición de una ventana mediante el proceso de diseño de la interacción. En la Tabla 1 se puede observar la clasificación de los elementos de iOS según [Apple, 2014a]. Con esto podemos apreciar la complejidad y alta variedad de opciones que ofrece la plataforma para la realización del diseño de la interacción.

Es preciso destacar que los nombres de los elementos no se van a traducir ya que en la guía de la plataforma están definidos en inglés y una traducción podría llevar a posibles confusiones a la hora de referenciarlos.

Tabla 1. Clasificación de elementos iOS

Categoría	Elementos
Bars	<ul style="list-style-type: none">- Status Bar- Navigation Bar- Toolbar- Tab Bar- Search Bar- Scope Bar
Content Views	<ul style="list-style-type: none">- Activity- Collection View- Image View- Map View- Page View Controller- Popover- Scroll View- Split View Controller- Table View- Text View- Web View
Controls	<ul style="list-style-type: none">- Activity Indicator- Contact Add Button- Data Picker- Detail Disclosure Button- Info Button- Label- Network activity indicator- Page Control- Picker- Progress View- Refresh Control- Rounded Rectangle Button- Segmented Control- Slider- Stepper- Switch- System Button

	- Text Field
Temporary Views	- Alert - Action Sheet - Modal View

2.2. Android

El sistema operativo Android apareció en el año 2008 y desde entonces ha tenido un crecimiento exponencial. El principal motivo de este crecimiento es el carácter abierto de este software, por lo que multitud de compañías tienen la posibilidad de acceder a él de forma gratuita y adaptarlo a sus nuevos productos, incluso para dar una imagen de marca personalizada y garantizar a sus usuarios una experiencia de usuario común que asocien a su marca. Desde entonces y debido principalmente a que multitud de compañías han optado por este producto, el sistema ha tenido que adaptarse y cambiar para poder dar soporte a por ejemplo los diferentes tamaños de pantalla que existen en el mercado entre otras capacidades hardware.

2.2.1. Evolución en el diseño de aplicaciones

La primera versión del sistema fue Android 1.0, la Figura 7 nos muestra como era el diseño original del sistema. Como se puede apreciar en esta primera versión el diseño era recargado, donde los iconos tenían mucho nivel de detalle.



Figura 7. Android 1.0 (tomada de [Cosmos, 2013])

Tras unos años continuando con un diseño muy similar al original, en Octubre del año 2011 aparece de forma oficial la versión 4.0 del sistema denominada Ice Cream Sandwich. Esta versión es el comienzo de un cambio generacional en el sistema que

nace de la necesidad de unificar dos versiones existentes hasta aquel momento, Android 2.3.x para smartphones y Android 3.x para tablets. Aprovechando esta unificación de dos sistemas existentes que conlleva la aparición de posibles aplicaciones universales, es decir, aplicaciones que se puedan ejecutar tanto en Smartphone como Tablet y que el diseño se adapte al tamaño de pantalla del dispositivo que la ejecuta, se intenta promover con mayor ahínco la Guía de diseño de aplicaciones de Android [Google, 2014a].

En la Figura 8 se muestra como pasó a ser el diseño del sistema a partir de la versión 4.0. Se puede apreciar a simple vista como los iconos han evolucionado mucho reduciendo el nivel de detalle e intentando ser mucho más simples. Lo mismo ocurre en la apariencia general de las aplicaciones que intentan cada vez ser más minimalistas y ofrecer al usuario sus diferentes opciones de forma simple y elegante.



Figura 8. Android 4.3 (tomada de [Cosmos, 2013])

Tras 3 años de trabajo en la unificación del sistema y varias versiones del mismo, el uso de éste creció muy rápido. Esto fue debido a la diversidad de marcas y dispositivos existentes en el mercado, que hacen que los usuarios dispongan de opciones que se adapten a sus necesidades y a su economía, pero también por la incursión de fabricantes chinos y la popularización del sistema en el país.

En junio de 2014 se presentó en una conferencia de desarrolladores protagonizada por Google denominada Google I/O la nueva versión de Android. Android 5.0 Lollipop trae como principal novedad el soporte a procesadores de 64bits junto a un renovado diseño denominado Material Design. Además de esto, a partir de este momento Android da el

salto oficial a los *wearables* mediante Android Wear, a los coches mediante Android Auto y a las televisiones mediante Android TV.

En el presente es relevante estudiar la nueva filosofía en el diseño de aplicaciones denominada Material Design. Dicha filosofía esta promovida por Google y como se destaca en [Google, 2014b] se caracteriza por centrarse en el usuario: “Material es una metáfora basada en la realidad táctil, inspirada en el estudio sobre el papel y la tinta, pero tecnológicamente hablando y abierto a la imaginación y a la magia” [Google, 2014b].

A continuación se procede al análisis de cómo esta nueva filosofía afecta al diseño de aplicaciones móviles, para ello observaremos diferentes ejemplos de evolución de aplicaciones antes de Android 5.0 y después.

En la Figura 9 se puede apreciar la evolución de la aplicación de correo que viene instalada en los dispositivos de la gama Nexus. En este caso se está comparando la barra lateral de menú o *drawer*, aquí se puede observar que en la imagen de la izquierda, Material Design, se han eliminado las líneas separadoras, se han espaciado los elementos y se le ha dado mayor relevancia a la selección de cuentas por parte del usuario en la parte superior.

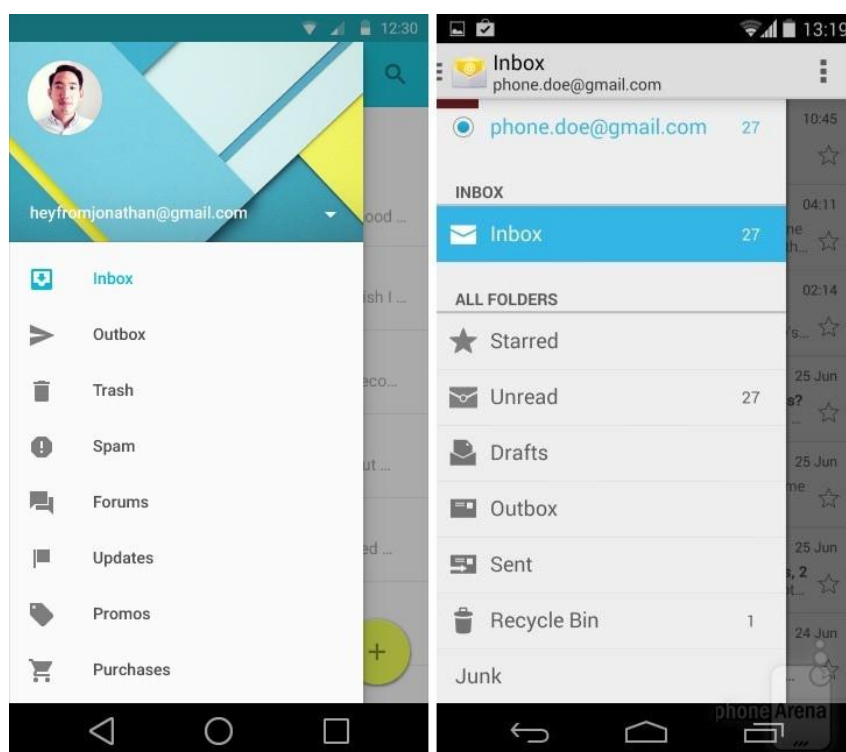


Figura 9. Evolución Material Design I (izquierda versión Material Design, derecha versión Android 4.4.4)

En la Figura 10 se analiza la evolución de la aplicación de correo Gmail que se puede descargar de la tienda de aplicaciones Google Play para el uso en cualquier dispositivo que disponga de Android. A la izquierda encontramos en nuevo diseño con Material Design y, como se puede observar, la interfaz ha sido simplificada. Para ello en la barra superior se ha eliminado el icono de la aplicación e iconos innecesarios. En cuanto al listado de mensajes, la información de cada uno de los mismos se ha espaciado, aunque este factor provoca que se puedan alojar menos mensajes en el mismo espacio, los mensajes son más fácil de leer. El hacer que todas las imágenes, junto a cada elemento, sean uniformes le proporciona estabilidad y coherencia al diseño.

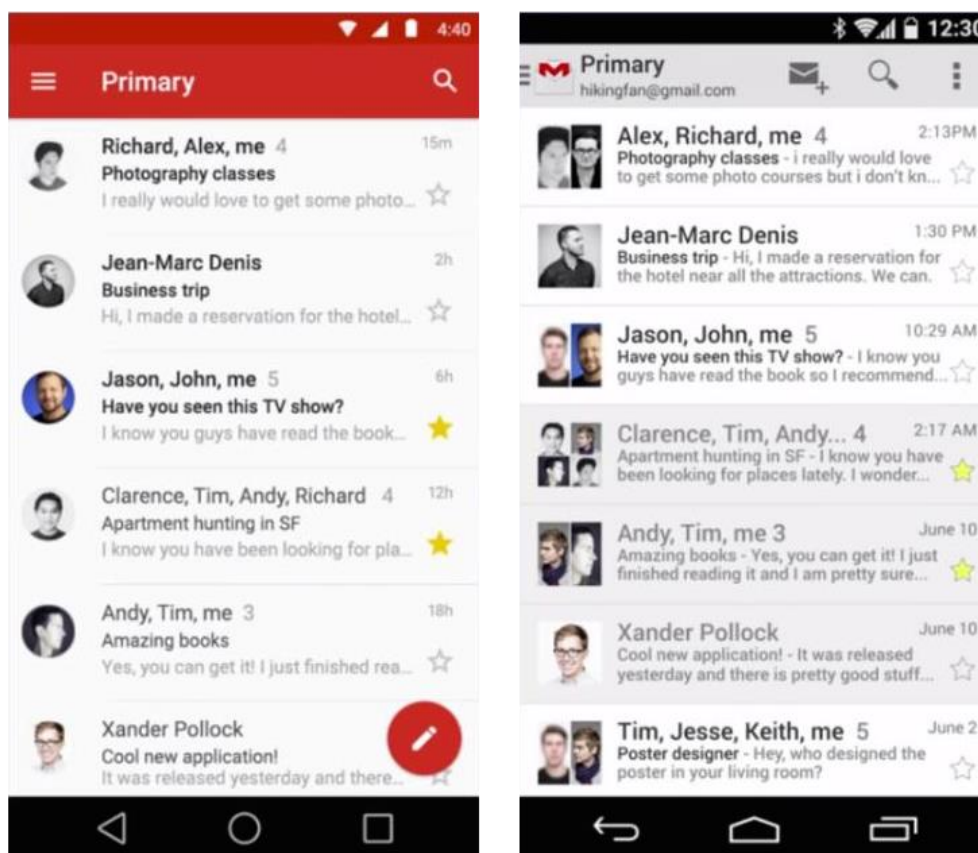


Figura 10. Evolución Material Design II (izquierda versión Material Design, derecha versión Android)

En las anteriores imágenes hemos podido comprobar como las diferencias entre la nueva y la anterior filosofía de diseño están enfocadas en el minimalismo, la simplicidad y la reducción de elementos o información innecesaria en la interfaz. En definitiva todos estos factores hacen que las técnicas de Material Design puedan tomarse como referente en el desarrollo de aplicaciones móviles.

2.2.2. Diversidad de versiones y capas de personalización

Al principio de este capítulo se mencionó que el sistema operativo Android es utilizado por una gran diversidad de compañías como por ejemplo: Samsung, HTC, LG o Motorola. Aunque también cabe destacar que en los últimos tiempos se han comenzado a popularizar compañías de origen chino como: Lenovo, Xiaomi, Meizu, Oppo, ZTE o Huawei.

Todas las compañías citadas tienen en común que en los dispositivos móviles que forman parte de su catálogo ofrecen el sistema operativo Android, pero a la hora de hacerlo le añaden modificaciones, aplicaciones propias y personalizaciones de diseño que hacen que los usuarios dispongan de una experiencia de usuario asociada a la

marca. Algunas compañías consideran estas modificaciones como una capa de personalización que añaden sobre el sistema operativo Android para aportar una diferencia significativa con respecto a sus competidores, como por ejemplo Samsung con Touchwiz o HTC con Sense. Sin embargo existen compañías que consideran que una modificación específica sobre el sistema es más que una simple capa de personalización y que lo que realmente aportan a sus usuarios es una versión nueva del sistema operativo Android, esto suele suceder con fabricantes de origen chino como puede ser Xiaomi con MIUI o Meizu con Flyme OS.

Tomando de base el enfoque de versión específica del sistema encontramos que en el mercado actual existen empresas como Cyanogen Inc, que ofrece una versión del sistema operativo Android denominada Cyanogenmod, que tiene sus propias características y los usuarios tienen la posibilidad de obtener este sistema a través de su web e instalarlo en sus propios dispositivos como alternativa al que les proporciona su fabricante.

Esta alta diversidad de opciones hacen que a veces los usuarios no estén habituados a las consideraciones que se detallan en las guías oficiales de Android ([Google, 2014a] y [Google, 2014b]) para el diseño de las interfaces de sus aplicaciones. Es común encontrar que determinados usuarios tienen interiorizados factores subjetivos asociados a la versión concreta de Android o capa de personalización que estén habituados a utilizar, considerando como habituales determinadas características que no están definidas en las guías oficiales de Android. Por ejemplo, en la Figura 11 se puede observar cómo la compañía Xiaomi (imagen de la izquierda) en la última versión del sistema operativo MIUI v6 se influencia más por las directrices de iOS 8 (imagen de la derecha) que por las propias de Android. Ofreciendo a sus usuarios mediante su versión de Android MIUI una experiencia que difiere de las consideraciones estándares de Android. Por todo esto, los diseñadores y desarrolladores que realizan aplicaciones para la plataforma Android tienen que analizar si quieren que su aplicación tenga un carácter global o que esté asociado de forma más concreta a la imagen de una determinada marca, de ser así habría que considerar las características propias de la misma. En el presente documento y debido al carácter global de las consideraciones declaradas en las guías oficiales de Android, se ha decidido centrar el análisis en dichas guías oficiales para poder realizar diseños de carácter generalista que abarquen un mayor número de usuarios, sin abordar versiones propietarias de Android.

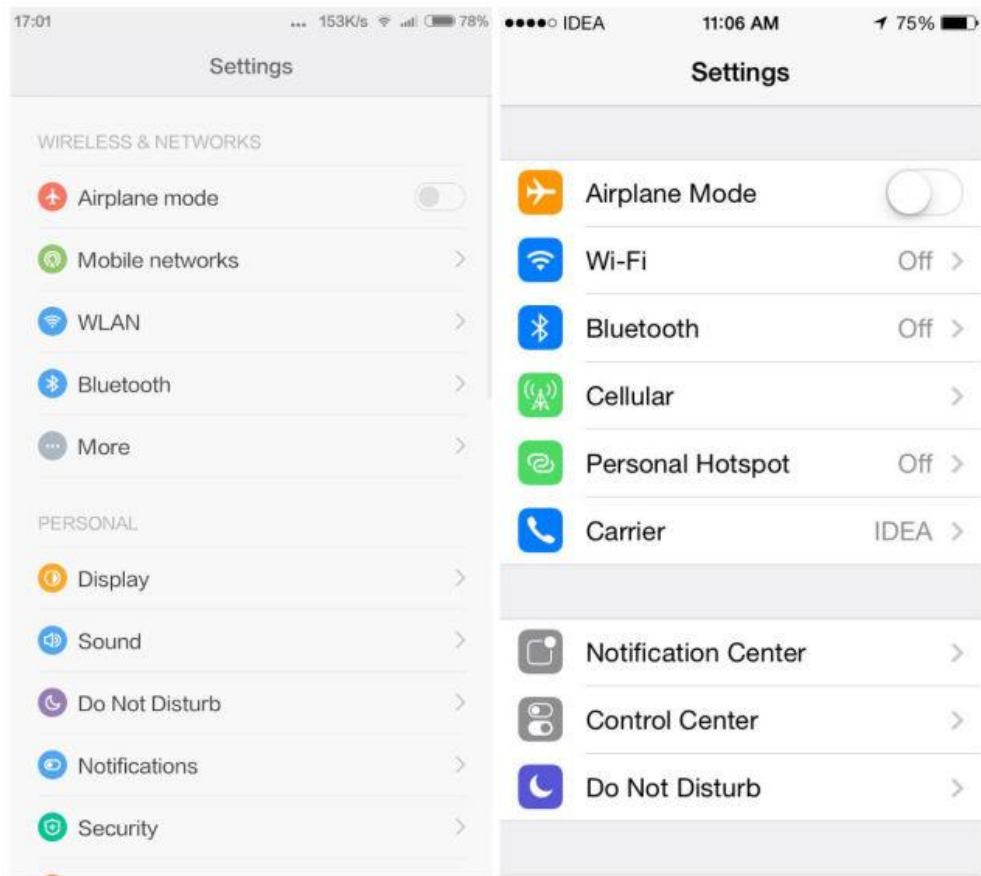


Figura 11. Miui v6 (izquierda) vs iOS 8 (derecha)

2.2.3. Filosofía de diseño

Al igual que se estudió en la parte de iOS, es necesario analizar la navegación entre los distintos niveles de una aplicación para la plataforma Android. El Action Bar, Figura 12, está situado en la parte superior de la ventana. En esta barra encontramos el icono o el nombre de la aplicación, un controlador para cambiar el contenido de la vista (sólo si contiene varias pantallas), botones que controlan acciones que pueda realizar el usuario y el botón de buscar si la aplicación soporta búsquedas. En niveles más profundos dentro de la navegación, al icono de la aplicación se le añade una flecha a la izquierda y entonces se convierte en el botón “arriba” (up), que una vez pulsado nos lleva a la vista raíz de la aplicación. El comportamiento estándar para volver a la ventana anteriormente visitada se realiza mediante la existencia de un botón físico de ir atrás (aparece en la parte inferior de la pantalla en aquellos dispositivos que no disponen de botón físico, Figura 13). Este botón consiste en, desde una vista, navegar a la vista justo anterior. Además no solo se limita a la vista anterior de la aplicación, sino que es un botón global en el contexto del dispositivo, es decir, que si el botón de atrás es pulsado en la vista raíz de la aplicación, la vista nos llevará a la anterior aplicación que tuviésemos abierta o en su defecto al “home” del dispositivo.

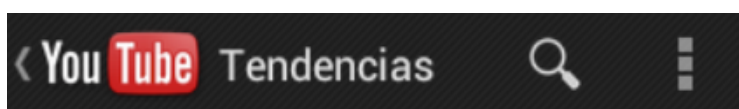


Figura 12. Action Bar Android



Figura 13. Botón atrás Android

En cuanto al análisis de los principios de diseño que forman parte de la filosofía de la plataforma, tanto las versiones de Android menor a 4.4 como las superiores contienen en sus guías de diseño unos principios que toda aplicación es recomendable que cumpla. A continuación, se analizarán los principios de diseño de los dos grandes grupos de Android.

Desde la versión 4.0 (Ice Cream Sandwich) hasta la 4.4.4 (Kitkat) del sistema Android podemos encontrar algunas claves para el correcto diseño de aplicaciones [Google, 2014a]:

- Los objetos reales son más divertidos que los botones y menús: permite a los usuarios tocar y manipular objetos en tu app. Esto reduce el esfuerzo cognitivo

necesario para completar una tarea además de hacerla emocionalmente satisfactoria.

- Llega a conocerme: aprende las preferencias de las personas a lo largo del tiempo en lugar de preguntarles.
- Hazlo breve: usa frases cortas con palabras simples, las personas suelen evitar las frases si son largas.
- Las imágenes son más rápidas que las palabras: considera el uso de imágenes para expresar ideas. Dichas imágenes atraen la atención de las personas y pueden ser mucho más eficientes que las palabras.
- Muéstrame lo que necesito cuando lo necesite: las personas se abruman cuando ven demasiada información. Por tanto oculta las opciones que no son necesarias en el momento y enséñales como llegar a ellas.
- Si tiene la misma apariencia, debe actuar de la misma forma: ayuda a las personas a identificar las diferentes funcionalidades haciendo que sean visualmente similares a otras aplicaciones.

A partir de la versión 5.0 (Lollipop) del sistema Android podemos encontrar algunas claves para el correcto diseño de aplicaciones [Google, 2014b]:

- Las superficies son intuitivas y naturales: las superficies y bordes proporcionan pistas visuales que se basan en nuestra experiencia con la realidad.
- Dimensiones que permiten la interacción: los fundamentos de la luz, superficie y movimiento son fundamentales para transmitir como interactúan los objetos.
- Un único diseño adaptativo: un único diseño organiza la interacción y el espacio. Cada dispositivo muestra una presentación diferente del mismo diseño, adaptando cada vista en tamaño e interacción apropiada para el dispositivo.
- El usuario inicia el cambio: los cambios en la interfaz son consecuencia de acciones del usuario.
- Acciones flotantes: es un trozo circular de “papel” separado del resto, el cual representa una única acción.

2.2.4. Elementos para el diseño de la interacción en Android

En esta sección, se van a identificar cuáles son los elementos que ofrece la plataforma para la composición de interfaces en el proceso de desarrollo de aplicaciones.

Al igual que cuando se detallaron los elementos de la plataforma iOS, no se va a proceder a traducir los nombres de estos elementos del inglés porque es así como aparecen en las guías de diseño de la plataforma.

En la Tabla 2 tenemos la clasificación de los elementos de Android según [Google, 2014a].

Tabla 2. Clasificación de elementos Android

Categoría	Elementos
Buttons	
Dialogs	<ul style="list-style-type: none">- Dialogs- Alerts- Popups- Toast
Grid Lists	<ul style="list-style-type: none">- Generic Grids- Grid List with Labels
Lists	
Patterns	<ul style="list-style-type: none">- Action Bar- Navigation Drawer- Multi-pane Layouts
Pickers	
Progress & Activity	<ul style="list-style-type: none">- Progress bars- Activity Indicators- Custom Indicators
Scrolling	<ul style="list-style-type: none">- Scroll Indicator- Index Scrolling
Seek Bars	
Spinners	<ul style="list-style-type: none">- Spinners in forms- Spinners in actions bars
Switches	<ul style="list-style-type: none">- Checkboxes- Radio Buttons- On/off Switches
Tabs	<ul style="list-style-type: none">- Scrollable Tabs- Fixed Tabs
Text Fields	

En la Tabla 3 tenemos la clasificación de los elementos de Android según [Google, 2014b].

Tabla 3. Clasificación de elementos Android Material

Categoría	Elementos
Bottom Sheets	
Buttons	<ul style="list-style-type: none"> - Flat & raised buttons - Floating action button - Icon Toggles - Dropdown buttons
Cards	
Chips	
Dialogs	<ul style="list-style-type: none"> - Alerts - Simple menus - Simple dialogs - Confirmation dialogs - Full-screen dialogs
Dividers	<ul style="list-style-type: none"> - Full-bleed dividers - Inset dividers - Subheaders and dividers
Grids	
List controls	<ul style="list-style-type: none"> - Checkbox - Switch - Reorder - Expand/collapse - Leave-behinds
Lists	
Menus	
Patterns	<ul style="list-style-type: none"> - Navigation Drawer
Pickers	<ul style="list-style-type: none"> - Data picker - Time
Progress & activity	<ul style="list-style-type: none"> - Linear - Circular
Sliders	<ul style="list-style-type: none"> - Continuous - Discrete
Snackbars & toasts	<ul style="list-style-type: none"> - Mobile Snackbar - Android toast
Subheaders	<ul style="list-style-type: none"> - List - Grid - Menu
Switches	<ul style="list-style-type: none"> - Checkbox - Radio Button - Switch
Tabs	<ul style="list-style-type: none"> - Fixed tabs - Scrollable tabs
Tooltips	

3. Desafíos del diseño multiplataforma

En los últimos siete años, desde que Apple presentó de forma oficial el primer iPhone, la evolución del mercado de smartphones ha sido exponencial. Han ido apareciendo de forma progresiva diferentes plataformas como Android o Windows Phone que compiten por ser los líderes en este mercado. Actualmente la situación de este mercado según un estudio del IDC sobre el uso de sistemas operativos en plataformas móviles [IDC, 2014], presentado en Agosto de 2014, indicaba que en el segundo cuatrimestre de 2014 las plataformas iOS y Android abarcaban un 96.4% del total de dispositivos vendidos en ese periodo, siendo este de 290.5 millones de dispositivos.

Tabla 4. Dispositivos por sistema operativo

Sistema Operativo	Total Dispositivos (millones)	Porcentaje
Android	255.3	84.7
iOS	35.2	11.7
Windows Phone	7.4	2.5
Blackberry	1.5	0.5
Otros	1.9	0.6
Total	301.3	100

Como podemos observar en la Tabla 4 la cantidad de dispositivos móviles que se venden en la actualidad cada cuatrimestre es muy alta y los sistemas operativos que dominan el mercado son Android e iOS. Por esta razón, en el presente trabajo éstas serán las plataformas que se estudiarán.

Como podemos apreciar y tras el gran dominio de ambas plataformas parece razonable que desarrolladores y compañías quieran publicar aplicaciones tanto en una como en la otra, para así poder abarcar prácticamente la totalidad de dispositivos móviles del mercado actual.

El desarrollo de aplicaciones para dos plataformas distintas, presenta unos desafíos, tanto técnicos como desde el punto de vista de la interacción, que se detallan en las dos siguientes subsecciones.

3.1. Punto de vista técnico

Desde un punto de vista técnico, evaluando tanto el sistema operativo necesario para desarrollar aplicaciones como los lenguajes y herramientas de programación podemos identificar ofertas diferentes según la plataforma para la que queramos desarrollar.

El desarrollo de aplicaciones para el sistema operativo Android se puede realizar sobre cualquiera de las principales plataformas de escritorio, Windows, Mac OS o Linux. En cuanto al lenguaje de programación que se utiliza es Java, y el entorno de desarrollo que se utilice es decisión de cada desarrollador. Habitualmente se suelen utilizar como

entornos de desarrollo Eclipse o Android Studio, una adaptación del conocido editor IntelliJ IDEA.

Para desarrollar aplicaciones sobre la plataforma iOS es requisito imprescindible disponer de un equipo con Mac OS, ya que esta es la única plataforma de escritorio sobre la que se pueden desarrollar aplicaciones iOS. En cuanto a los lenguajes de programación, se utilizan Objective-C y Swift, que son los lenguajes utilizados tanto para el desarrollo de aplicaciones iOS como de Mac OS. El entorno de desarrollo que se utiliza es Xcode.

Podemos observar en la Tabla 5 un resumen de la plataforma, lenguaje de programación y entorno de desarrollo necesarios para desarrollar aplicaciones iOS y Android.

Tabla 5. Desarrollo aplicaciones

Sistema Operativo	Plataforma necesaria	Lenguaje de programación	Entorno de desarrollo
iOS	Mac OS	Objective-C, Swift	Xcode
Android	Windows, Mac OS, Linux	Java	Eclipse, Android Studio

Como se puede apreciar, existen notables diferencias en el desarrollo de aplicaciones para ambas plataformas. Dichas diferencias no sólo aparecen en las plataformas de desarrollo o lenguajes de programación sino también están presentes en el uso de las librerías propias asociadas a los diferentes lenguajes, el uso de los elementos disponible por cada plataforma o la gestión de operaciones multihilo. Todas estas circunstancias han llevado a la proliferación, en los últimos años, de entornos de desarrollo que permiten generar aplicaciones para ambas plataformas móviles. De este modo el proceso de desarrollo se simplifica ya que los desarrolladores trabajan en un único entorno y con un lenguaje determinado para luego generar de forma sencilla una versión de la aplicación diferente para cada plataforma.

Algunas de las herramientas más populares para el desarrollo de aplicaciones multiplataforma son PhoneGap, Titanium Appcelerator o Intel XDK, las cuales tienen diferentes características a la hora de desarrollar las aplicaciones y generar el código en las diferentes plataformas [Angulo, 2014a].

3.2. Punto de vista del diseño de la interacción

Desde el punto de vista del diseño de la interacción las plataformas iOS y Android comparten similitudes y diferencias. Centrándonos en el campo de desarrollo de aplicaciones tanto iOS como Android podemos apreciar que ambas plataformas disponen de guías para el desarrollo de interfaces ([Apple, 2014a] y [Google, 2014a] respectivamente) en las que se ven reflejados diferentes aspectos, como pueden ser las filosofías de diseño de la plataforma, la normativa de accesibilidad correspondiente y el

uso de los diferentes elementos proporcionados por la plataforma para la construcción de interfaces.

Para el desarrollo de aplicaciones multiplataforma es necesario tener en cuenta las consideraciones de diseño de cada plataforma, así como los factores asociados a la interacción que hacen que un usuario se sienta cómodo en el uso de aplicaciones de una plataforma concreta. En el capítulo 2 se ha podido observar como existen diferencias en el diseño de la interacción de aplicaciones para dos plataformas diferentes que hacen que un usuario tenga un conocimiento general de la composición típica de las ventanas de las aplicaciones y del uso y distribución de los elementos que ofrece su plataforma por las diferentes ventanas de las aplicaciones. Por todo esto, podemos apreciar la existencia de diferentes estrategias de diseño multiplataforma, como se cita en [Barea, 2013], el desarrollo de aplicaciones multiplataforma puede abordarse desde tres posibles aproximaciones: Dependiente de la Plataforma, Multiplataforma Propia y Multiplataforma Adaptada. Cada una de estas aproximaciones tiene un impacto diferente sobre el nivel en el que las aplicaciones sigue las recomendaciones de las guías de diseño de la plataforma para la que está desarrollada, siendo la opción Dependiente de la Plataforma la que sigue de forma más fiel las recomendaciones de las guías, seguido de Multiplataforma Adaptada que pretende proporcionar una imagen de marca en ambas plataformas pero adaptando la interacción al cumplimiento de las guías. Por último, tenemos la opción Multiplataforma Propia, que se centra en mantener una imagen de marca común a ambas plataformas sin seguir de forma estricta las recomendaciones de las guías.

El presente documento se centrará en la comparación de las guías de desarrollo de aplicaciones de iOS y Android para que los diseñadores y desarrolladores puedan tener una visión de conjunto de cuales son los elementos que tienen disponibles en ambas plataformas, con sus similitudes y diferencias de uso, además de proporcionarles una idea de cómo abordar el diseño de la interacción multiplataforma de forma sencilla, facilitándoles la elección de las herramientas de desarrollo multiplataforma [Angulo, 2014a] o nativas y la toma de decisiones de estrategias de diseño [Barea, 2013]. En especial, el presente trabajo será de utilidad cuando se opte por una estrategia Dependiente de la Plataforma o Multiplataforma Adaptada.

4. Marco de diseño de la interacción en aplicaciones móviles: Navegación y distribución de elementos

Cuando se quiere abordar el diseño de la interacción en aplicaciones móviles es necesario tener en cuenta en primer lugar la estructura que se quiere emplear según los niveles de navegación de la aplicación y la distribución de elementos dentro de cada ventana de la aplicación.

4.1. Niveles de navegación

Durante el estudio de las guías de diseño de las plataformas iOS y Android se encontró necesario el estudio de los niveles de navegación presentes en una aplicación. Para ello se ha tomado como punto de partida lo mencionado al respecto en [Google, 2014a], realizando un análisis en profundidad de dichos niveles de navegación. Estos niveles son superior o raíz, medio y detalle.

En la Figura 14 podemos observar un ejemplo del mapa de navegación de la aplicación de la Escuela en la que se pueden observar de forma clara como las ventanas de la misma se distribuyen en: nivel superior (en azul), nivel medio (en verde) y nivel detalle (en naranja).

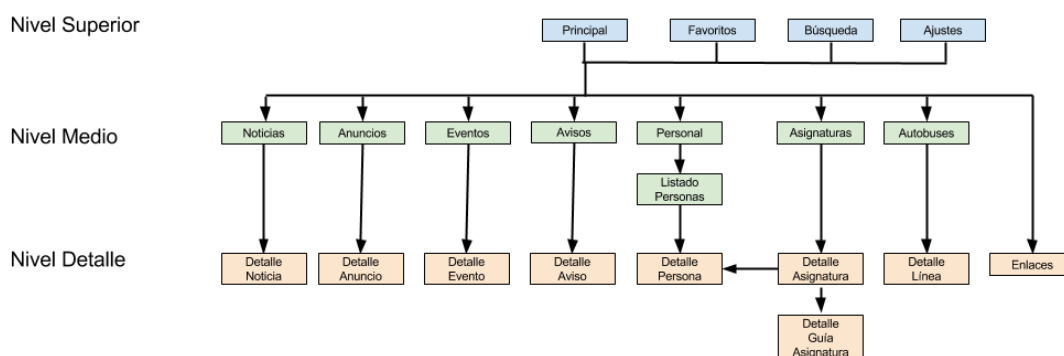


Figura 14. Ejemplo de niveles de navegación

- Nivel Superior o raíz: este nivel es el que nos encontramos al abrir una aplicación, se caracteriza por mostrar al usuario de una forma rápida y visual las diferentes secciones que componen la aplicación para que sea capaz de comprender de forma rápida la estructura global de la misma.

En la Figura 15 podemos apreciar el nivel superior de la aplicación oficial de la Escuela, donde el usuario puede moverse entre las cuatro secciones principales que componen este nivel: ETSINF, Favoritos, Búsqueda y Perfil.

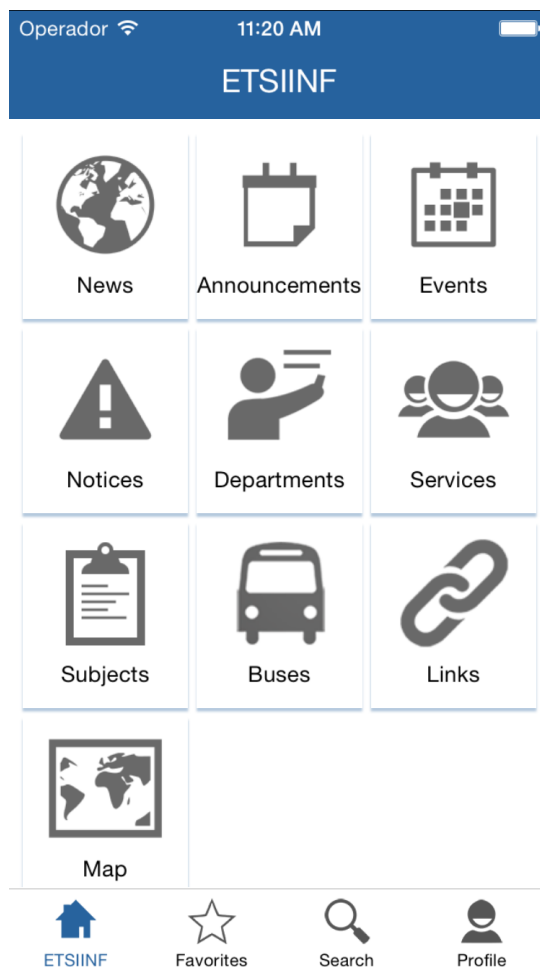


Figura 15. Nivel superior app ETSIINF

- Nivel Medio: este nivel se caracteriza por servir de transición o puente entre el nivel superior y el de detalle, su finalidad es mostrar un listado de elementos u opciones para que el usuario puede seleccionar de una forma simple de qué elemento desea obtener más información.

En la Figura 16 se puede observar un claro ejemplo de nivel medio sobre la aplicación de la Escuela, en este caso el usuario ha pulsado la opción Noticias del Nivel Superior y ahora ha pasado a encontrarse frente a un listado de todas las noticias que hay disponibles actualmente en la Escuela. El usuario podrá seleccionar en este listado el elemento que sea de su interés para obtener una información detallada del mismo.

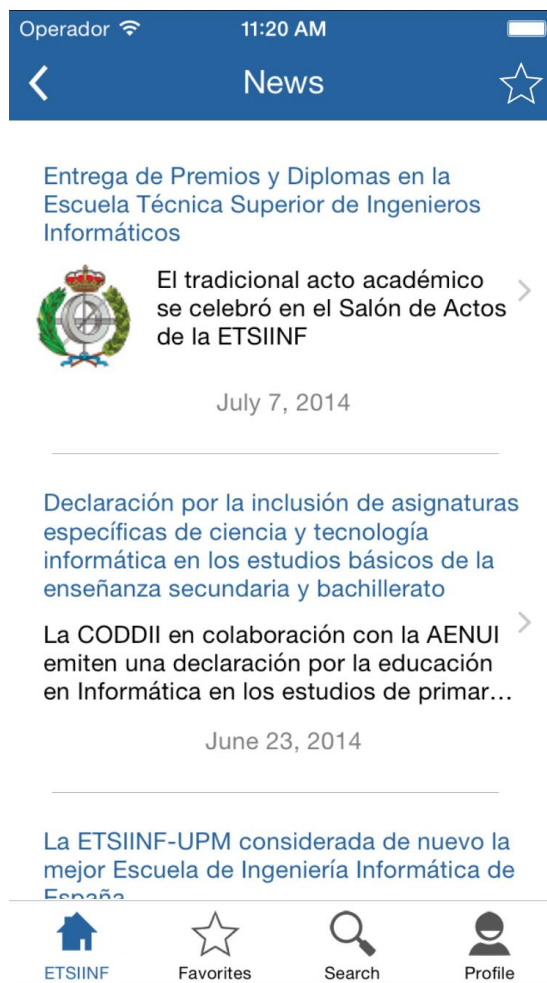


Figura 16. Nivel Medio app ETSIINF

- Nivel Detalle: este nivel sirve para mostrar información detallada de un elemento concreto de la aplicación. Se caracteriza por disponer de elementos que permitan interactuar con la información mostrada en este nivel como por ejemplo poder compartir el contenido en redes sociales.

En la Figura 17 encontramos un ejemplo del nivel de detalle, en este caso es una noticia concreta, seleccionada del listado de noticias visualizado en el nivel medio.



Figura 17. Nivel Detalle app ETSIINF

4.2. Distribución de elementos dentro de la ventana

Los elementos que componen la ventana de una aplicación se distribuyen a lo largo de la misma en base a una organización determinada en cada una de las plataformas que estamos analizando.

En la Figura 18 podemos observar la composición estándar de una ventana de una aplicación en la plataforma iOS. Dicha ventana se organiza en las siguientes tres zonas:

- Zona de navegación: en ella se ubica la barra de navegación que servirá como punto de referencia al usuario, para poder identificar de forma sencilla en qué ventana de la aplicación se encuentra en un momento determinado. Está compuesta por 3 áreas que se distinguen a simple vista, a la izquierda encontramos el botón de atrás, el cual nos llevará a la ventana anterior, en el centro encontramos de forma habitual el nombre de la ventana actual, y en la parte derecha se pueden encontrar de forma opcional botones que realicen alguna acción. La implementación de la barra de navegación es necesaria únicamente en aquellas aplicaciones que requieran navegación entre distintos subniveles.
- Zona de contenido: en ella se localiza el contenido principal de la ventana. Dicho contenido debe cumplir las consideraciones y restricciones indicadas en [Apple, 2014a] sobre usabilidad y accesibilidad, haciendo un uso correcto de los diferentes elementos proporcionados por el sistema.
- Zona de herramientas: en esta zona podemos encontrar de forma habitual un *Tab Bar* que servirá para poder movernos por diferentes secciones de una misma aplicación. También es habitual encontrar un *Toolbar* o barra de acciones como se puede observar en la Figura 18. La implementación de esta zona es opcional y quedará a la elección de los diseñadores o desarrolladores en base a cómo se quiera organizar la información dentro de la aplicación.

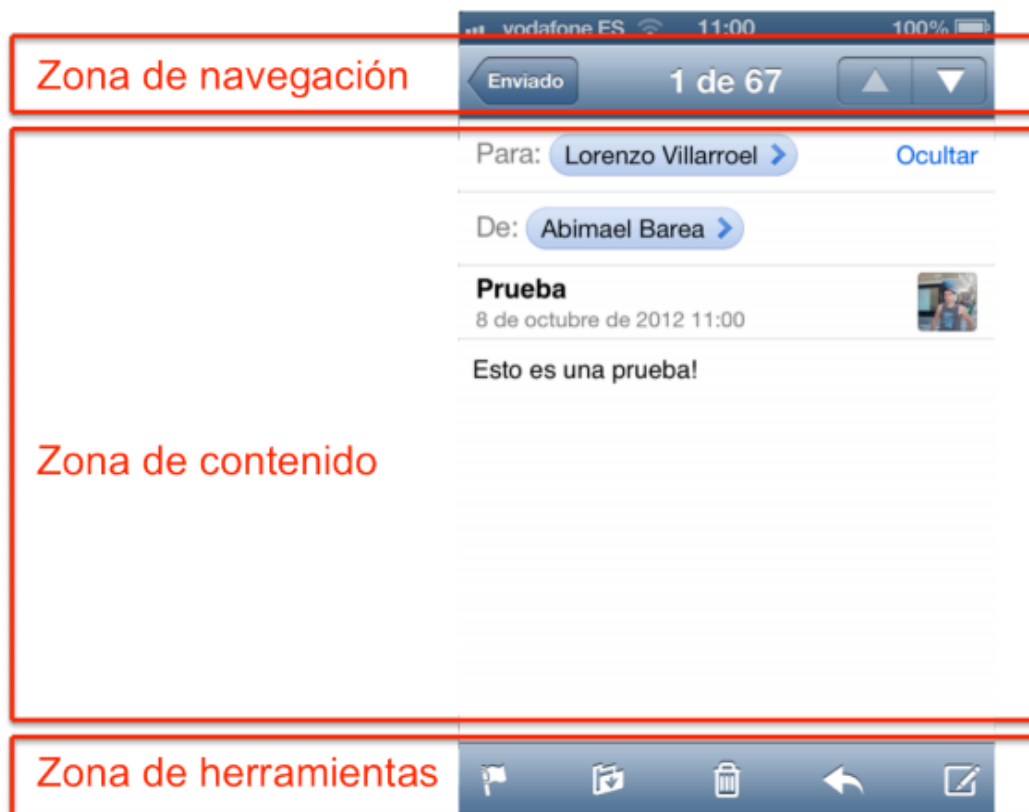


Figura 18. Distribución ventana iOS

Las aplicaciones Android, a diferencia de las de iOS, sólo disponen de dos zonas principales, acción y contenido, aunque en la zona de contenido existe un mayor número de combinaciones a la hora de distribuir la información:

- **Zona de acción:** en ella se sitúa el *Action Bar*. Dicha barra se divide en una imagen con el icono de la aplicación situada en la parte izquierda de la barra. A continuación existe la posibilidad de incluir un controlador de vista (*spinner* o *drawer*) o el nombre identificativo de la ventana en la que nos encontramos. En la parte derecha de la barra solemos encontrar botones de acción, si el número de estos es muy alto y el dispositivo no dispone de un botón de acción físico podemos encontrar un botón denominado Action Overflow que tendrá la apariencia de 3 cuadrados apilados de forma vertical y el cual nos mostrará opciones adicionales.
- **Zona de contenido:** en la Figura 19 se puede observar que la zona de contenido Android puede dividirse en varias partes, que son: barra superior, contenido principal y barra inferior. La implementación de estas barras es algo opcional y se ajustará a las necesidades del problema concreto que se esté desarrollando.

- Barra superior: en esta zona se suele encontrar de forma habitual elementos relacionados con la organización del contenido principal, como por ejemplo *tab bars* o *spinners*.
- Contenido principal: el contenido ubicado en esta zona debe cumplir las consideraciones y restricciones indicadas en [Google, 2014a] y [Google, 2014b] sobre usabilidad y accesibilidad, haciendo un uso correcto de los diferentes elementos proporcionados por el sistema.
- Barra inferior: esta parte de la ventana suele ir relacionada con acciones que realizan modificaciones sobre la parte de contenido principal o como extensión de la zona de acción de la aplicación para destacar determinadas acciones en vez de situarlas en la parte derecha de dicha barra.

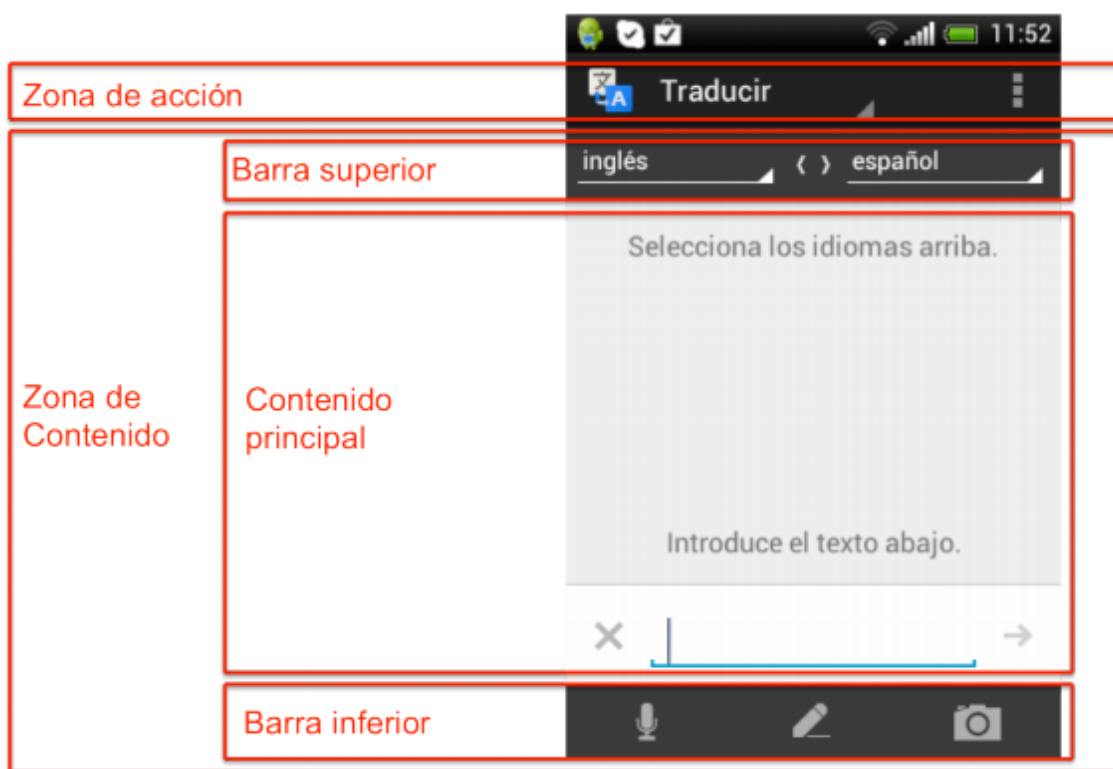


Figura 19. Distribución ventana Android

5. Marco de diseño de la interacción en aplicaciones móviles: Criterios de clasificación

En el presente capítulo se va a realizar una clasificación propia de los elementos que se utilizan para el desarrollo de interfaces en las diferentes plataformas. Para ello se han tomado como base las clasificaciones de los elementos en las diferentes plataformas, analizadas en el capítulo 2. Además, se han realizado las siguientes tareas:

1. Definir categorías: al no existir una definición clara de categorías de elementos en la plataforma Android, se han tomado como punto de partida las categorías de elementos presentes en la plataforma iOS: Bars, Content Views, Controls y Temporary Views. Sobre estas categorías se ha decidido añadir dos más para que la clasificación sea más clara: Navigation y Progress & Activity.
2. Eliminar elementos: tras la observación de los elementos presentes en las diferentes plataformas se ha decidido centrarnos, únicamente, en el análisis y clasificación de aquellos que sirven para estructurar la información en el nivel superior de navegación, los que se utilizan en el nivel medio y los que sirven para dar feedback o proporcionar acciones básicas a los usuarios. Otros elementos propios del nivel de detalle como podrían ser: Image View, Map View o Web View se han eliminado para hacer la clasificación más sencilla, puesto que al seleccionar los elementos que componen la ventana del nivel de detalle existen un amplio grado de libertad y dependerá del contenido que muestre la aplicación concreta.
3. Identificar elementos equivalentes: se han identificado las equivalencias entre los elementos de las diferentes plataformas, ya que en cada una elementos similares aparecen en la mayoría de los casos con una nomenclatura diferente en cada plataforma.

Tras realizar estas tareas, se procedió a la elaboración de la comparativa, en las dos siguientes subsecciones se analizarán en detalle los elementos que forman parte de esta comparativa para una mayor comprensión de los mismos.

5.1. Comparativa de elementos

En la Tabla 6 se detalla la clasificación realizada de los elementos por categorías y plataformas, y las equivalencias identificadas entre los elementos de las diferentes plataformas. Navigation se ha establecido como primera categoría en la clasificación porque se considera que la navegación entre los diferentes apartados de la aplicación es una actividad que corresponde al nivel superior de navegación descrito en el capítulo anterior y es conveniente abordarla, por esta razón, en primer lugar.

Tabla 6. Clasificación unificada de elementos de interacción

Categoría	iOS	Android	Android Material
Navigation	Tab Bar	Tab Scrollable Tab Fixed Tab	Tab Scrollable Tab Fixed Tab
	Scope bar	Spinner / Scrollable Tab	Menu / Scrollable Tab
		Navigation Drawer	Navigation Drawer
	Split View Controller	Multi-pane Layout	
Bars	Status bar	Status bar	Status bar
	Navigation Bar	Action Bar	App Bar
	Toolbar	Split Action Bar	Toolbar
Content Views	Table View	List	List
	Collection View	Grid List	Grid
Controls	Data Picker	Picker	Picker
	Slider	Seek Bar	Continuous/Discrete Slider
	Switch	Switch Checkboxes Radio Buttons On/off Switches	Switch Checkboxes Radio Buttons On/off Switches
	Segmented Control	Spinner / Scrollable Tab	Menu / Scrollable Tab
	Detail Disclosure Button		
	Page Control		
	Stepper		
Progress & Activity	Activity Indicator/ Network activity indicator	Activity indicator	Linear/circular indicator
	Progress View	Progress bar	
	Refresh Control		
		Custom indicator	
Temporary Views	Alert	Alert	Alert
	Action Sheet	Dialog	Dialog simple/confirmation
	Activity	Popup	Bottom Sheet
		Toast	Android Toast
			Mobile snackbar
	Modal View		Full Screen Dialog

5.2. Análisis de elementos

Tras haber llevado a cabo la clasificación de los elementos presentes en el marco se procede a analizar los elementos presentes en el mismo según las categorías identificadas.

5.2.1. Navigation

En esta categoría se analizan elementos que sirven para estructurar el contenido de las aplicaciones en el nivel superior de navegación de las mismas.

Tab Bar vs Tab vs Tab

Un Tab Bar sirve para cambiar el contenido de la ventana. En iOS se encuentra en la parte inferior de la misma y no puede tener más de 5 tabs visibles. En android el Fixed Tab se coloca debajo del Action Bar o App Bar y suele tener 3 elementos máximo si se encuentra en el nivel superior de navegación. Por el contrario, si está en el nivel medio puede tener más de 3 elementos y lo denominaríamos Scrollable Tab.

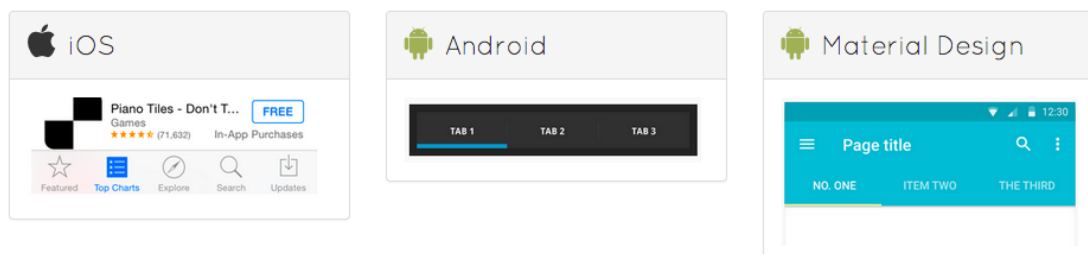


Figura 20. Ejemplos de Tab Bars

Scope Bar vs Spinner vs Nenu

La Scope Bar, en iOS, sirve para cambiar o filtrar el contenido que se está visualizando en la ventana actual, al igual que sus equivalentes, el Spinner y el Menu, en Android.

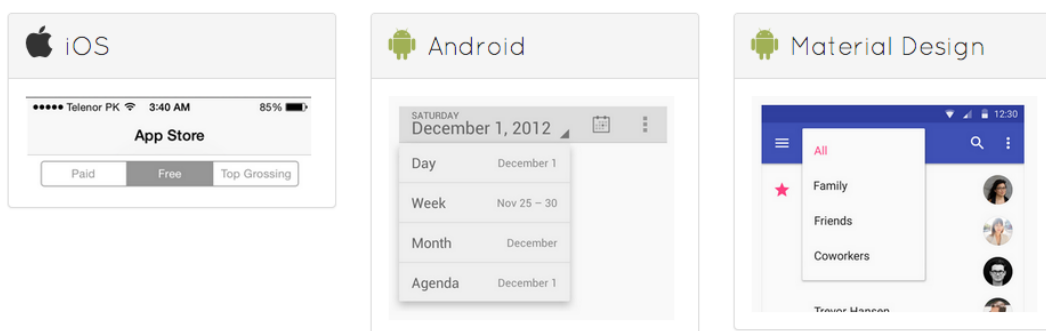


Figura 21. Ejemplos de Scope Bar, Spinner y Menu

Navigation Drawer

Este elemento sólo está presente en el sistema operativo Android y sirve como sustituto del Tab en el nivel superior de navegación cuando se muestran en él más de 4 elementos principales.

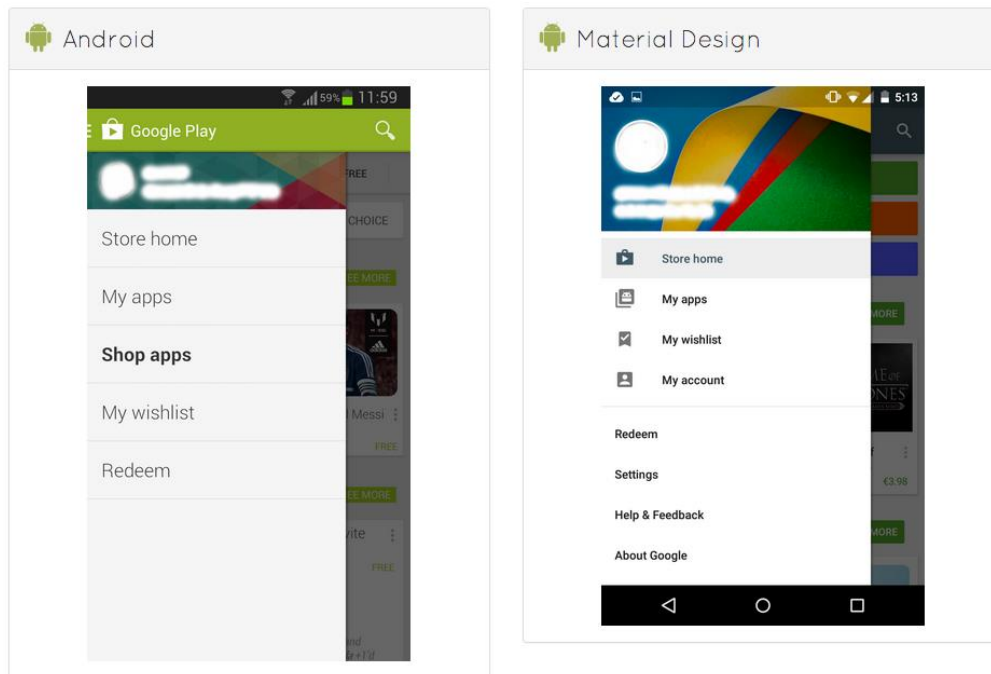


Figura 22. Ejemplos de Drawer

Split View Controller vs Multi-Pane Layout

Estos elementos son usados como patrón en el diseño de aplicaciones y consiste en tener un listado de opciones que hace de menú y una ventana con detalle de cada elemento del menú, en smartphones se representa en dos ventanas y en tablets en una.

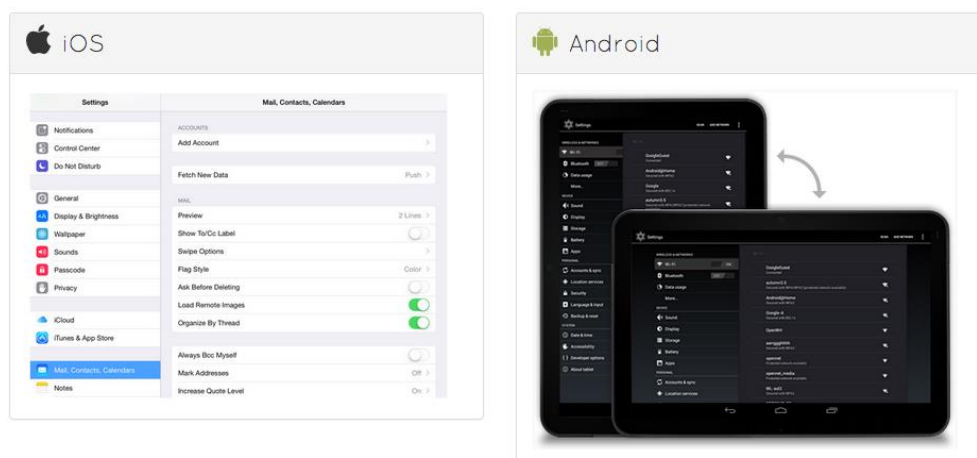


Figura 23. Ejemplos de Split View y Multi-Pane Layout

5.2.2. Bars

La presente categoría trata de analizar los diferentes tipos de barras que se pueden encontrar en las ventanas que componen una aplicación móvil.

Status Bar vs Status Bar vs Status Bar

La Status Bar está situada en la parte superior de las aplicaciones, es controlada por el sistema y muestra información del mismo, como por ejemplo el estado de la batería. Esta barra puede ser ocultada por los programadores cuando quieren que la aplicación ejecute a pantalla completa, como por ejemplo en los juegos.

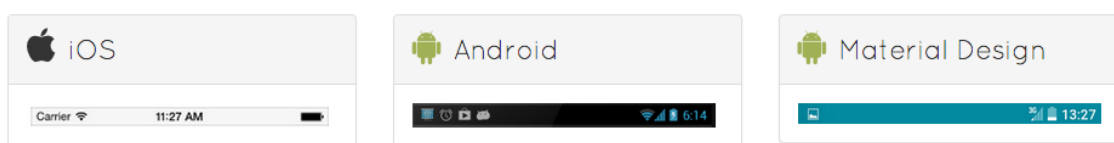


Figura 24. Ejemplos de Status Bar

Navigation Bar vs Action Bar vs App Bar

Estas barras se sitúan en la parte superior de la aplicación, correspondiendo a la Zona de Navegación en iOS y de Acción en Android. Su finalidad es mostrar el título de la ventana actual y acciones asociadas a la misma. El usar estas barras es opcional.

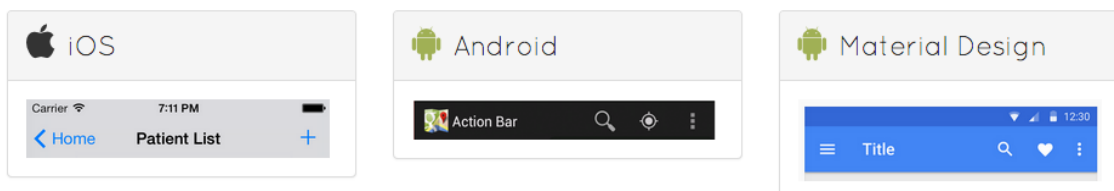


Figura 25. Ejemplos de Navigation Bar, Action Bar y App Bar

Toolbar vs Split Action Bar vs Toolbar

Estas barras sirven para albergar opciones sobre una ventana concreta, suelen usarse cuando el número de opciones es alto y no se desea sobrecargar las Navigation, Action o App Bar o cuando se le quiere dar importancia a las mismas. Tradicionalmente se sitúan en la parte inferior de la ventana.

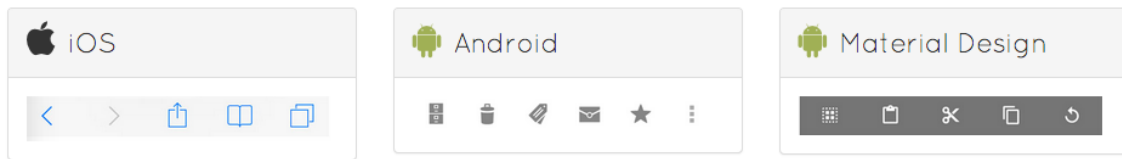


Figura 26. Ejemplos de Toolbar y Split Action Bar

5.2.3. Content Views

Esta categoría engloba elementos de tipo contenedor que sirven para mostrar listados o conjuntos de elementos de forma agrupada.

Table View vs List vs List

Estos elementos son usados, generalmente, en el Nivel Medio de Navegación de una aplicación. Se usan para mostrar listados de elementos, que pueden ser, por ejemplo: conversaciones, contactos, aplicaciones, etc.

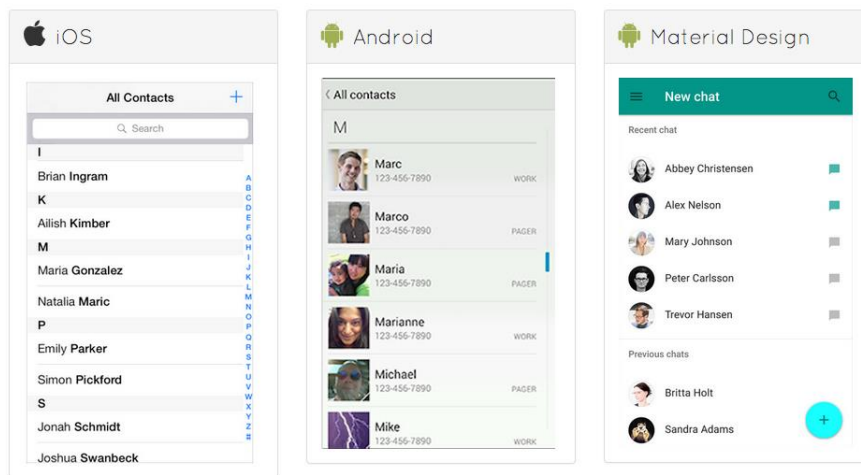


Figura 27. Ejemplos de Table View y Lists

Collection View vs Grid List vs Grid

Estos elementos se usan para mostrar listados de objetos, de forma más visual. Acompañados, normalmente, por imágenes o iconos.

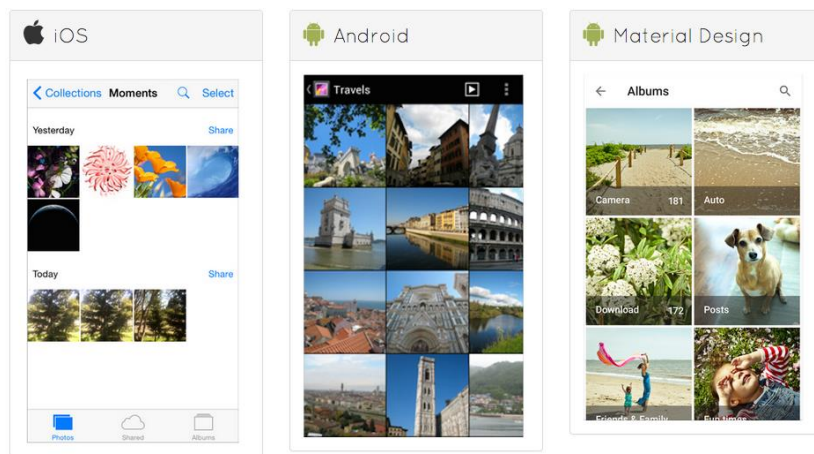


Figura 28. Ejemplos de Collection View y Grids

5.2.4. Controls

En esta categoría se analizan elementos que sirven para realizar acciones sobre el contenido de la aplicación.

Data Picker vs Picker vs Picker

Estos elementos proporcionan la posibilidad de seleccionar un único valor entre un conjunto acotado de valores.

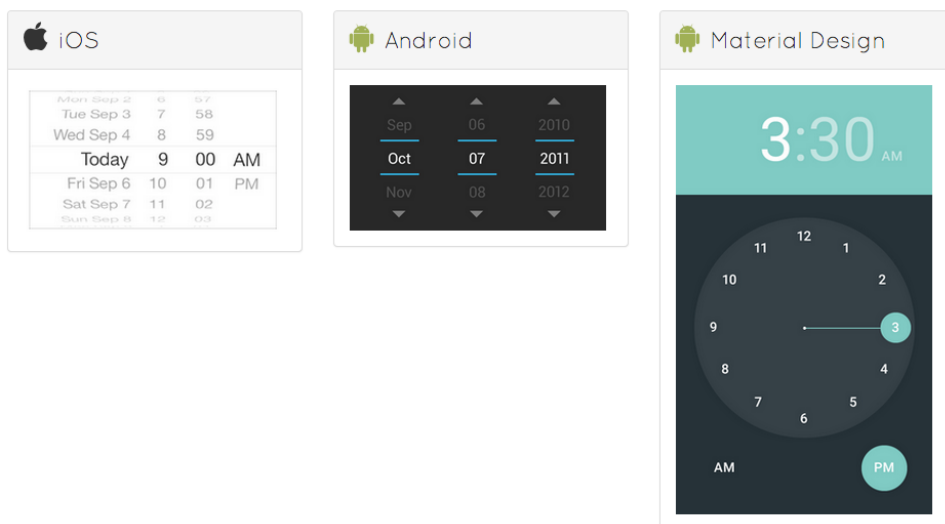


Figura 29. Ejemplos de Data Picker y Pickers

Slider vs Seek Bar vs Continuous/Discrete Slider

Estos elementos dan la opción de seleccionar un valor de un conjunto continuo o discreto, moviendo el indicador con el dedo a lo largo de la barra.

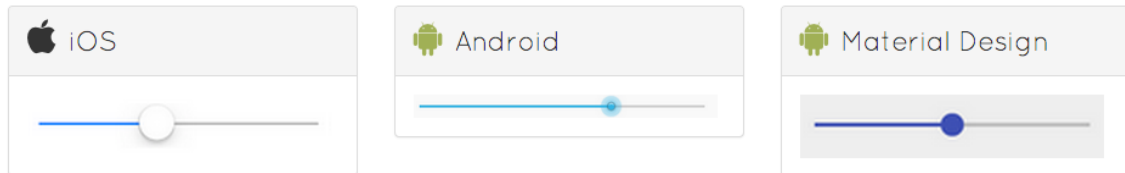


Figura 30. Ejemplos de Slider, Seek Bar y Continuous/Discrete Slider

Switch vs Switch vs Switch

Estos elementos presentan dos opciones o estados posibles.

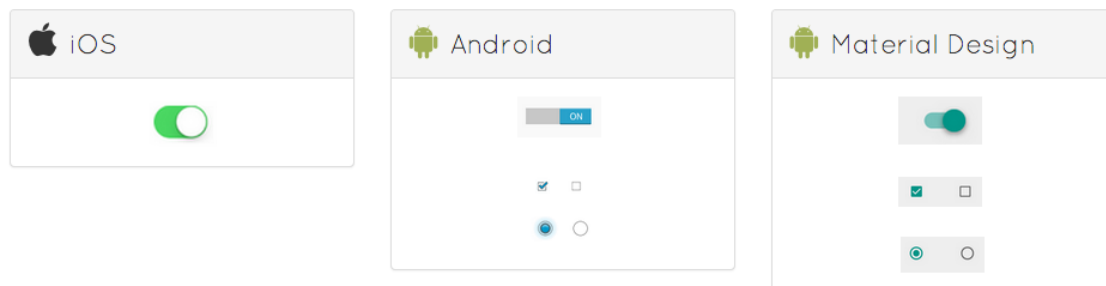


Figura 31. Ejemplos de Switch

Segmented Control vs Spinner/Scrollable Tab vs Menu/Scrollable Tab

Estos elementos proporcionan la posibilidad de cambiar el contenido visualizado en la ventana actual, aplicando sobre el mismo diferentes filtros o modificaciones.

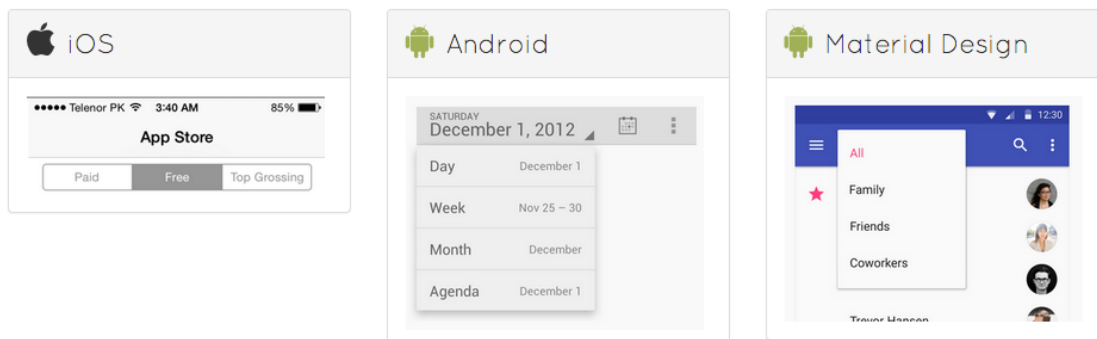


Figura 32. Ejemplos de Segmented Control, Spinner y Menu

Detail Disclosure Button

Muestra detalles adicionales o funcionalidades concretas asociadas a un elemento determinado. Habitualmente se suele usar en Table Views.



Figura 33. Ejemplo de Detail Disclosure Indicator

Page Control

Sirve para indicar el número de vistas disponibles en una ventana, además de mostrar en cuál de ellas se encuentra el usuario en un momento determinado.

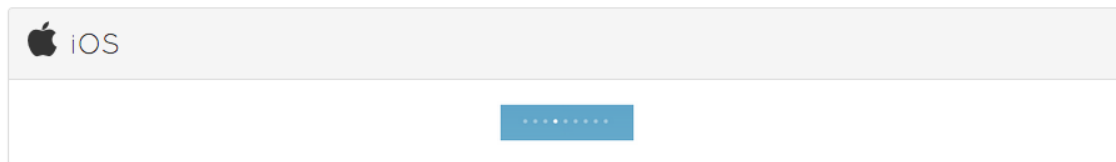


Figura 34. Ejemplo de Page Control

Stepper

Se utiliza para aumentar o disminuir un valor en un tamaño constante.

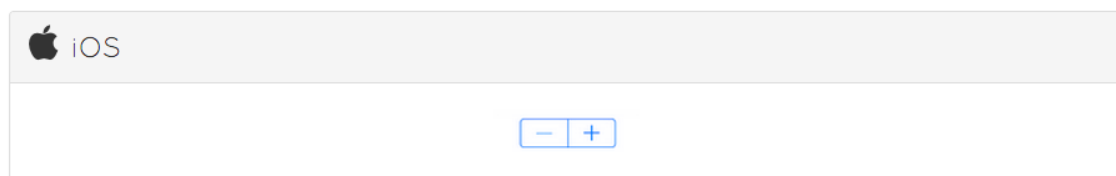


Figura 35. Ejemplo de Stepper

5.2.5. Progress & Activity

En esta categoría se analizan elementos encargados de proporcionar feedback al usuario cuando la aplicación realiza tareas que pueden conllevar un tiempo prolongado.

Activity Indicator/ Network activity indicator vs Activity Indicator vs Linear/Circular indicator

Este tipo de indicadores son especialmente útiles para proporcionar información al usuario sobre que se está realizando una operación que conlleva un tiempo de espera.

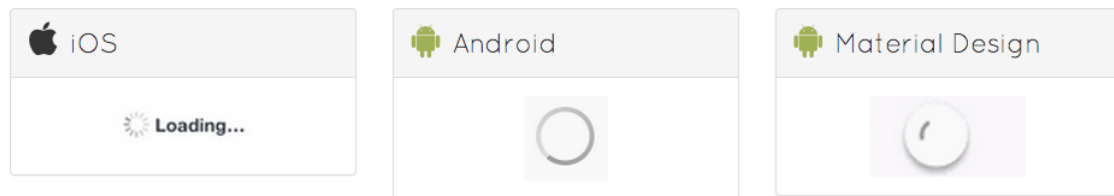


Figura 36. Ejemplos de Activities Indicators

Progress View vs Progress Bar

Estos elementos sirven para dar información al usuario sobre el progreso de una tarea que conlleva un tiempo de espera, como por ejemplo la descarga de un fichero o la instalación de una nueva aplicación.

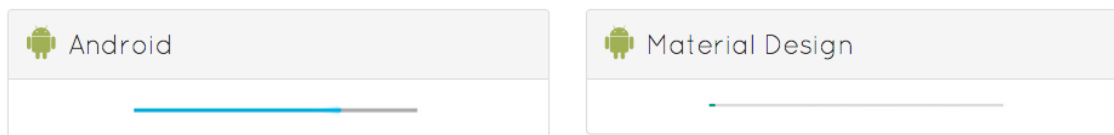


Figura 37. Ejemplos de Progress View y Progress Bar

Refresh Control

Este elemento permite a los usuario, poder recargar el contenido de una tabla o listado de elementos, tirando del listado hacia abajo hacia abajo.

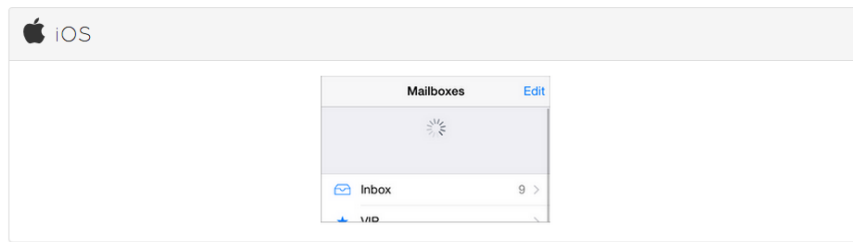


Figura 38. Ejemplo de Refresh Control

Custom Indicator

Este elemento sirve para mostrar progreso de tareas que tardan un tiempo en completarse, con la característica de que el programador o diseñador puede personalizar su apariencia en base a sus necesidades.

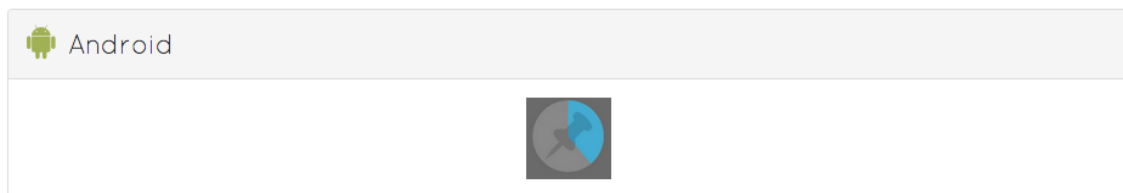


Figura 39. Ejemplo de Custom Indicator

5.2.6. Temporary Views

Esta categoría abarca a elementos que tienen un carácter temporal, es decir, que aparecen en el contexto de la aplicación cuando se les requiere y que tras su uso desaparecen.

Alert vs Alert vs Alert

Los Alerts sirven para informar al usuario sobre una situación que requiere de su conocimiento o confirmación para proceder.

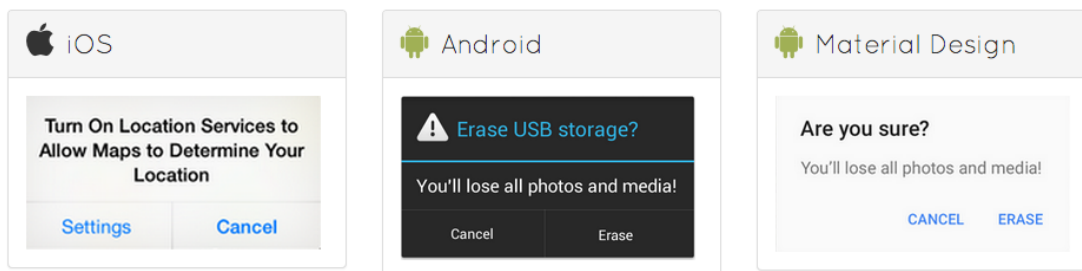


Figura 40. Ejemplos de Alerts

Action Sheet vs Dialog vs Dialog simple/confirmation

Estos elementos muestran al usuario información adicional o necesaria para tomar una decisión.

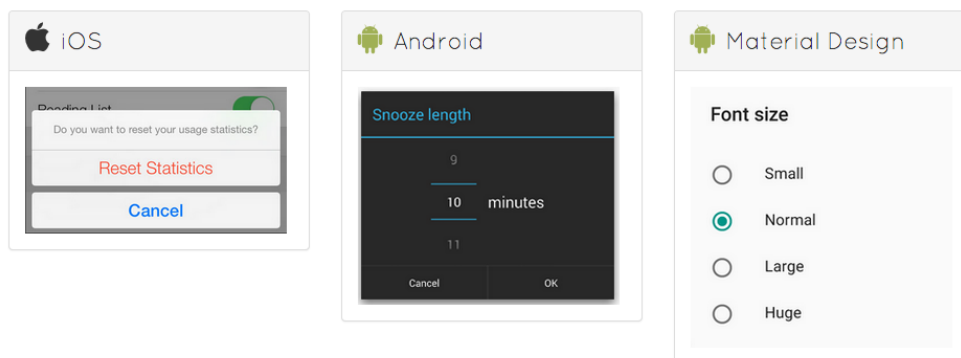


Figura 41. Ejemplos de ActionSheet y Dialogs

Activity vs Popup vs Bottom Sheet

Presentan al usuario una serie de opciones, que requieren una única selección de entre las disponibles.

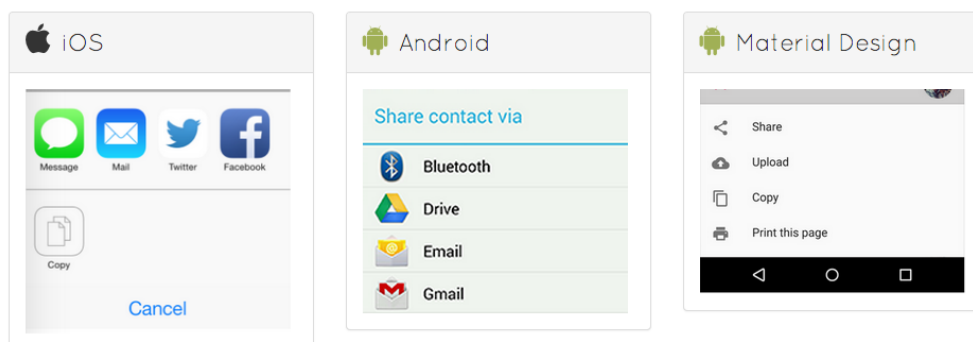


Figura 42. Ejemplos de Activity, Popup y Bottom Sheet

Toast vs Android Toast

Presenta información al usuario sobre una acción determinada que se acaba de realizar, tras unos segundos desaparece de forma automática.

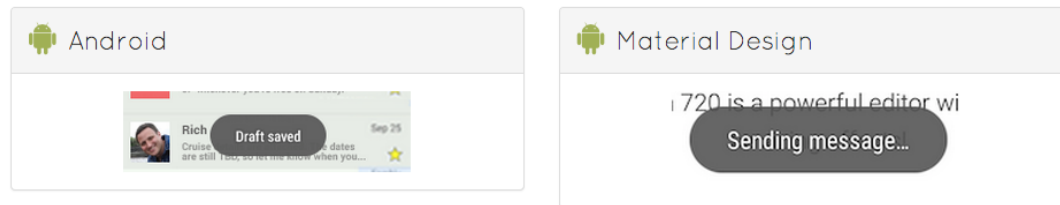


Figura 43. Ejemplos de Toast

Mobile Snackbar

Al igual que los Toast, proveen información al usuario sobre una acción determinada que acaba de suceder, pero con la peculiaridad de que pueden contener acciones, como por ejemplo deshacer.

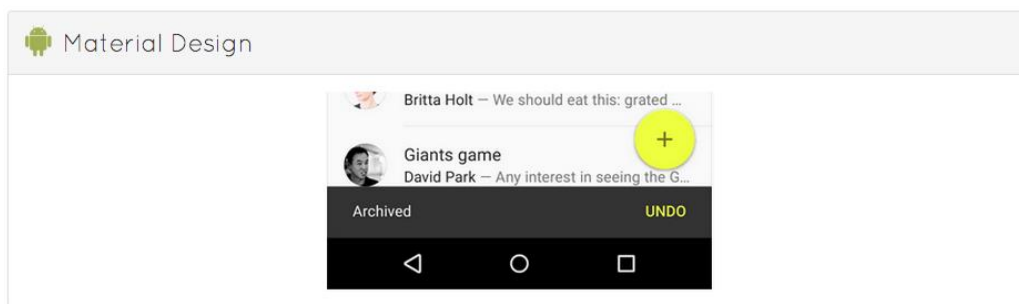


Figura 44. Ejemplo de Mobile Snackbar

Modal View vs Full Screen Dialog

Son ventanas que proveen funcionalidades determinadas dentro del contexto de la tarea o actividad que se está realizando.

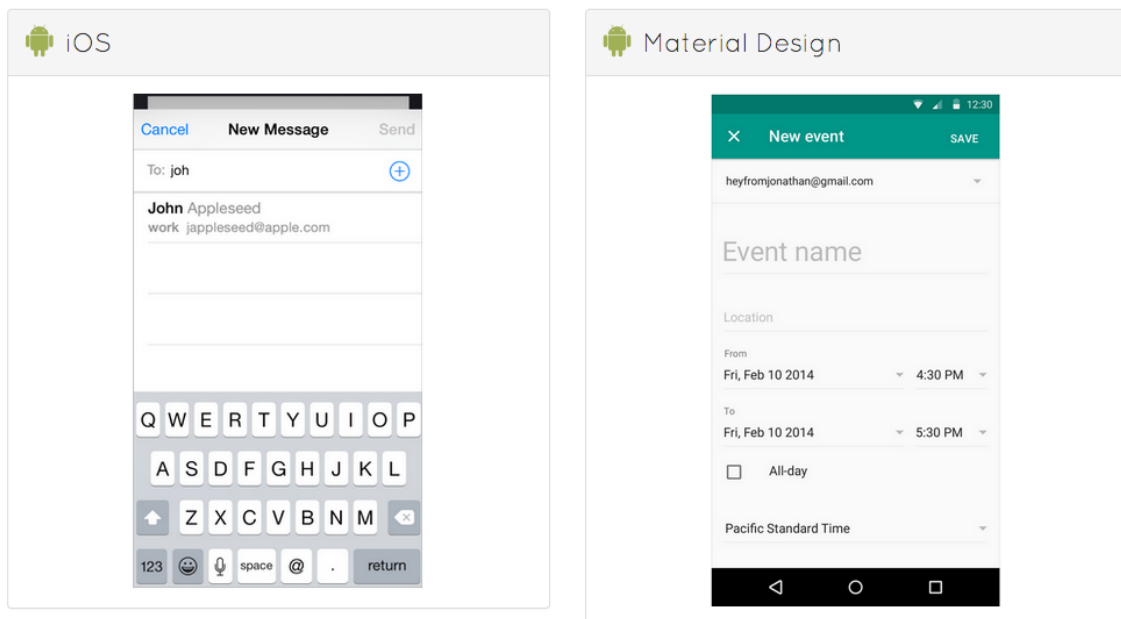


Figura 45. Ejemplos de Modal View y Screen Dialog

6. Caso de estudio

Para evaluar la viabilidad del marco de diseño de la interacción para aplicaciones móviles se detalla en el presente capítulo como se han aplicado los contenidos del mismo en el desarrollo de una aplicación móvil real. La aplicación que va a servir como caso de estudio es la aplicación oficial de la Escuela, la cual está siendo desarrollada en el Laboratorio de Ingeniería del Software y en cuyo desarrollo el autor del presente trabajo ha tenido un papel destacado.

En primer lugar se va a realizar un análisis de la estructura de la aplicación partiendo de los diferentes niveles de navegación presentes en la misma, a continuación se estudiará cuál es la distribución de los elementos de una ventana concreta.

Una vez realizado el análisis estructural de la aplicación, se observarán los elementos usados para componer la interfaz en ambas plataformas en base a la clasificación desarrollada en el capítulo 5.

6.1. Estructura de la aplicación

La estructura de la aplicación cumple con los niveles de navegación definidos por el marco. En la Figura 46 se encuentra el mapa de navegación de la aplicación en el que se puede apreciar de forma visual como existen los tres niveles correspondientes a los definidos en el marco: Nivel Superior, Nivel Medio y Nivel Detalle.

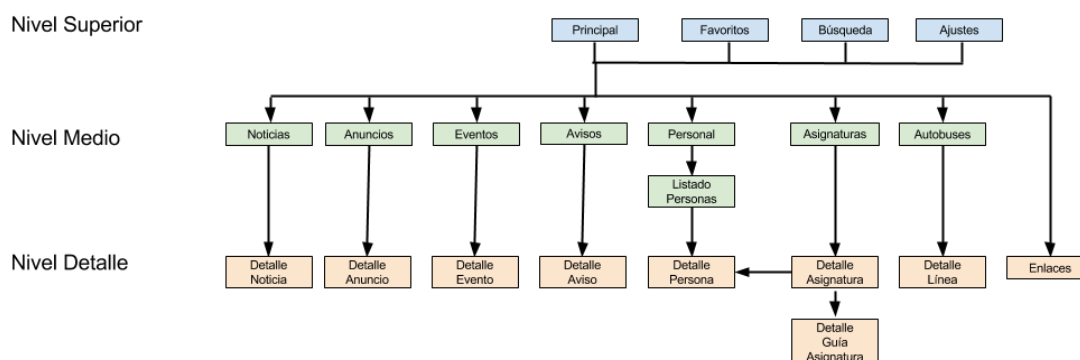


Figura 46. Mapa de Navegación App Escuela

En la Tabla 7 observamos las secciones representadas en el mapa de navegación asociados a su nivel de navegación correspondiente.

Tabla 7 Niveles App Escuela

Nivel Superior	Nivel Medio	Nivel Detalle
Principal	Noticias	Detalle Noticia
Principal	Anuncios	Detalle Anuncio
Principal	Eventos	Detalle Evento
Principal	Avisos	Detalle Aviso
Principal	Personal/Listado Personas	Detalle Persona
Principal	Asignaturas	Detalle Asignatura / Detalle Guía
Principal	Autobuses	Detalle Línea
Principal	-	Enlaces
Favoritos	-	-
Búsqueda	-	-
Ajustes	-	-

Estudiando la estructura de una ventana concreta de la versión iOS de la aplicación podemos ver como, en la Figura 47, se identifican de forma evidente las zonas definidas por el marco.

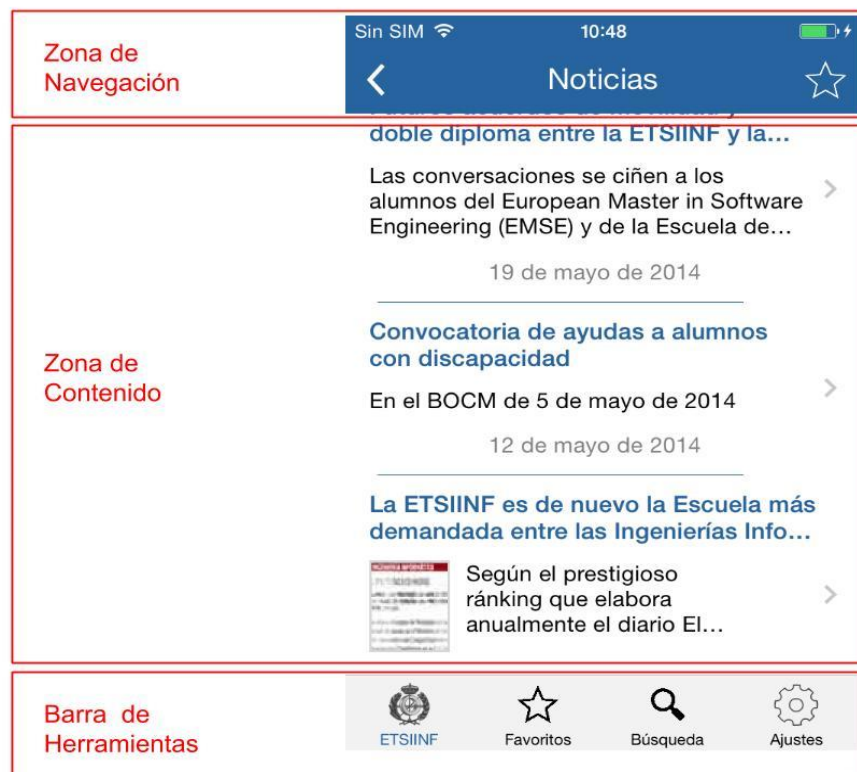


Figura 47. Estructura App en iOS

De forma análoga a como se ha observado con la versión de iOS, en la correspondiente versión Android de la aplicación podemos ver como, en la Figura 48 se identifican las zonas definidas por el marco.



Figura 48. Estructura App en Android

Observando la misma ventana en ambas versiones de la aplicación se puede identificar como en las dos existe una zona superior en la que se sitúan acciones sobre la ventana además del nombre de la misma, en iOS denominada Zona de Navegación y en Android Zona de Acción. Seguidamente, se puede identificar como en ambas versiones existe una Zona de Contenido. Para terminar, en iOS la Barra de Herramientas está visible en todas las secciones de la aplicación y permite navegar entre los niveles superiores de navegación presentes en la aplicación. En la versión Android no se ofrece esta opción, pero sí la posibilidad de cambiar el contenido visualizado en la Zona de Contenido mediante una barra superior (Scrollable Tab) situada en dicha Zona.

6.2. Análisis de elementos

La aplicación de la Escuela ha sido desarrollada únicamente para iOS y Android, aún no ha sido adaptada al diseño Material Design de Android. Por esto, en la Tabla 8, se aprecian en primer lugar, resaltado en color naranja, los elementos de los que hace uso la aplicación para las plataformas iOS y Android. A continuación, el marco nos permite determinar de forma sencilla, resaltado en color verde, los elementos equivalentes que habría que utilizar durante el desarrollo de una futura versión de Material Design.

Tabla 8. Resumen Elementos App Escuela

Categoría	iOS	Android	Android Material
Navigation	Tab Bar	Tab Scrollable Tab Fixed Tab	Tab Scrollable Tab Fixed Tab
	Scope bar	Spinner	Menu
		Navigation Drawer	Navigation Drawer
	Split View Controller	Multi-pane Layout	
Bars	Status bar	Status bar	Status bar
	Navigation Bar	Action Bar	App Bar
	Toolbar	Split Action Bar	Toolbar
Content Views	Table View	List	List
	Collection View	Grid List	Grid
Controls	Data Picker	Picker	Picker
	Slider	Seek Bar	Continuous/Discrete Slider
	Switch	Switch Checkboxes Radio Buttons On/off Switches	Switch Checkboxes Radio Buttons On/off Switches
	Segmented Control	Spinner / Scrollable Tab	Menu / Scrollable Tab
	Detail Disclosure Button		
	Page Control		
	Stepper		
Progress & Activity	Activity Indicator/ Network activity indicator	Activity indicator	Linear/circular indicator
	Progress View	Progress bar	
	Refresh Control		
		Custom indicator	
Temporary Views	Alert	Alert	Alert
	Action Sheet	Dialog	Dialog simple/confirmation
	Activity	Popup	Bottom Sheet
		Toast	Android Toast
			Mobile snackbar
	Modal View		Full Screen Dialog

En la Figura 49 se puede observar la ventana que permite visualizar enlaces de interés relativos a la Escuela o a la Universidad. Como se puede observar en la parte superior de la Zona de Contenido de ambas versiones, se sitúa un elemento que permite filtrar el contenido mostrado en la ventana, en Android (a la izquierda) se utiliza un Scrollable Tab mientras que en iOS (a la derecha) se utiliza un Segmented Control. Ambos elementos son equivalentes como se puede observar en la sección de Controls del marco (ver Tabla 6). Estos elementos tienen la finalidad de clasificar y filtrar el contenido visualizado en la ventana que se encuentren. Debajo de ese elemento aparece el listado de enlaces que en Android se visualiza mediante el elemento List, mientras que en iOS se utiliza para dicha finalidad un Table View. Estos elementos son análogos, como se observa en la sección Content Views del marco.



Figura 49. Comparación de elementos en Android e iOS

7. Herramienta web

Para facilitar la aplicación del marco que se detalla en el presente trabajo se ha decidido realizar una aplicación web, debido a que las tecnologías web son multiplataforma y esto simplifica el proceso de difusión. De esta forma, para dar a conocer el marco es necesario distribuir únicamente el enlace de la aplicación web.

En las subsecciones posteriores, se analizarán las tecnologías web utilizadas en el desarrollo de la herramienta y la estructura y diseño de la misma.

7.1. Tecnologías web

Las tecnologías de referencia en el desarrollo de páginas y aplicaciones web son HTML y CSS. A continuación se explicará cuales son las versiones que se van a utilizar de ambas tecnologías y las características de las mismas.

HTML5 es el último estándar presente para HTML. La versión anterior de HTML fue la 4.01, la cual había sido publicada en el año 1999. HTML5 fue diseñado para reemplazar tecnologías como HTML4 o XHTML, ya que fue especialmente pensado para llevar contenido enriquecido sin la necesidad de usar *plugins* adicionales. La versión actual es capaz de soportar diferente tipo de contenidos, desde animaciones hasta gráficos pasando por música, vídeo o incluso poder desarrollar aplicaciones web complejas.

Algunas de las interesantes nuevas características que podemos encontrar en HTML5, son las siguientes [W3C, 2012a]:

- Las etiquetas <video> y <audio> que permiten dar soporte a elementos multimedia.
- La etiqueta <canvas> para poder dibujar elementos en 2D.
- Nuevas etiquetas para clasificar el contenido de la web como son por ejemplo: <header>, <nav> o <footer>.
- Soporte a almacenamiento local.

CSS3 es el último estándar para CSS, es compatible con versiones anteriores de CSS gracias a que ha sido dividido en módulos. Esa versión contiene por un lado la antigua especificación de CSS, dividida en pequeñas piezas, y por otra parte la incorporación de nuevos módulos, descritos en [W3C, 2012b], como por ejemplo:

- Layouts de multiples columnas.
- Animaciones.
- Efectos para textos.
- Transformaciones 2D y 3D.

7.1.1. Responsive Design

El diseño web adaptable, conocido en inglés como Responsive Web Design, se trata de una filosofía de diseño y desarrollo que consiste en adaptar la apariencia de las páginas web al dispositivo que se esté utilizando para visualizarla según explica [Quesada, 2013].

En la actualidad existen multitud de dispositivos con diferentes características y con esta tecnología se pretende que un único diseño sea capaz de visualizarse de forma adecuada en cada dispositivo.

Este tipo de diseño se ha popularizado mucho en los últimos años gracias a que usando tecnologías como HTML5 y CSS3 es mucho más sencillo para los desarrolladores llevar a cabo este tipo de diseños, garantizando a su vez una experiencia de usuario de alta calidad.

Actualmente este tipo de diseño es muy popular ya que gracias a él es posible mantener una imagen corporativa multiplataforma unificada. Para llevarlo a cabo de forma sencilla y estandarizada existen diferentes frameworks que ayudan en el proceso de realizar aplicaciones web conformes a esta filosofía. Por ejemplo, Google, en su última conferencia de desarrolladores (Google I/O 2014), presentó “Material Design” filosofía con la que tratan de establecer unas directrices en el diseño multiplataforma [Google, 2014b].

7.1.2. Análisis de frameworks para Responsive Design

Los frameworks que ayudan a los desarrolladores web en el uso del diseño adaptativo son muy variados, a continuación nos centraremos en analizar tres de las alternativas más populares, basándonos en el estudio realizado por [Haveri, 2014].

Las alternativas que contemplamos son Bootstrap, Foundation y Skeleton. En Tabla 9 tenemos una breve comparativa del número de columnas y ejemplos de sintaxis de cada una de las tecnologías.

Tabla 9. Comparativa frameworks Responsive Design

Nombre	Licencia	Empresa	Nº columnas	Ejemplos
Bootstrap 3	MIT License	Twitter	12	.col-xs-1
Foundation v5	MIT License	ZURB	12 defecto (1-16)	.small-1.columns .medium-1.columns
Skeleton	MIT License	Dave Gamache	16	.one.column

Pese a las grandes similitudes de los tres frameworks se ha decidido utilizar Bootstrap porque es especialmente útil para el desarrollo de prototipos rápidos. Otro de los motivos que han llevado a esta elección es que existe una gran comunidad de desarrollo detrás y además existía un conocimiento previo de este framework.

7.2. Diseño de la herramienta

La herramienta web desarrollada para visualizar la información desarrollada en el marco de diseño de aplicaciones, descrito en el presente trabajo, consiste en una prueba de concepto que intenta acercar el marco a través de un medio visual y multiplataforma como es la web.

Para ello, la herramienta ha sido alojada en un servidor de la Escuela y se puede acceder a la misma a través de la siguiente dirección web:

<http://raptor.ls.etsiinf.upm.es/marcoappsmoveiles/>

7.2.1. Estructura

La herramienta se compone de cuatro secciones principales, que son descritas a continuación:

- Principal: esta sección contiene una información resumida del contenido que se puede encontrar dentro de la herramienta y hace de página de bienvenida de la misma.
- Estructura: en esta sección encontraremos la información relativa a la estructuración de aplicaciones móviles definidas por el marco de diseño.
- Elementos: en esta sección tendremos de forma visual una comparativa entre los elementos utilizados para el diseño de la interacción a través de las diferentes plataformas.
- Acerca De: esta sección contiene información sobre los autores de la herramienta, referencias bibliográficas e información de contacto sobre soporte de la herramienta.

En la Figura 50, en la que se visualiza la sección principal de la herramienta, se puede observar como en la parte superior derecha de la herramienta está presente un menú que nos permite navegar entre las diferentes secciones descritas anteriormente.



Marco de diseño tiene como objetivos:

- Facilitar la toma de decisiones de diseño de la interacción para aplicaciones móviles
- Analizar la estructura de las aplicaciones móviles en las diferentes plataformas
- Comparar las diferencias y similitudes entre los elementos usados en el diseño de la interacción entre ambas plataformas

Figura 50. Secciones de la herramienta

7.2.2. Caso de estudio

En esta subsección se analizará como se puede utilizar la herramienta para realizar el diseño de la interacción de una aplicación. Para ello abordaremos las siguientes decisiones que habría que tomar al inicio de un nuevo proyecto, con sugerencias de dónde se puede encontrar información y capturas de la herramienta:

- Decisión de la estructura y secciones de la aplicación: la herramienta podría ayudar, si acudimos a la sección estructura y miramos como se organizan las aplicaciones desde un punto de vista de los niveles de navegación de la misma.



Figura 51. Apartado Estructura (1 de 2) de la herramienta

- Distribución del contenido de una ventana: del mismo modo que para tomar decisiones sobre la estructura de las secciones de la aplicación, en la sección Estructura podemos comprender cuáles son las regiones o zonas que existen para distribuir el contenido dentro de cada ventana de las aplicaciones móviles.



Figura 52. Apartado Estructura (2 de 2) de la herramienta

- Selección de elementos a utilizar y equivalencias en otras plataformas: la sección elementos de la herramienta es especialmente útil para la comprensión de los elementos disponibles en cada una de las plataformas y las equivalencias entre los mismos.



Figura 53. Apartado Elementos de la herramienta

8. Conclusiones

Durante la realización del presente trabajo se han alcanzados todos los objetivos descritos en la propuesta de trabajo desarrollada por el tutor. Estos objetivos fueron:

- Estudiar las diferencias en diseño de la interacción entre las plataformas Android e iOS, tanto para teléfonos móviles como para tabletas.
- Diseño de un marco de diseño de la interacción para aplicaciones móviles.
- Aplicación del marco elaborado a un caso de estudio de aplicación multiplataforma en Android e iOS.
- Evaluación de aplicaciones móviles según el marco propuesto.

Además de alcanzar los objetivos perseguidos, durante el desarrollo del mismo se ha profundizado en el desarrollo de aplicaciones móviles multiplataforma, sacando a la luz y relacionando aspectos de interés que no se habían publicado anteriormente.

El alto número de dispositivos móviles presentes en el mercado en la actualidad y la gran diversidad de marcas que ofrecen este tipo de dispositivos han propiciado una gran evolución en el diseño de aplicaciones móviles en un corto espacio de tiempo, haciendo que la complejidad del diseño de aplicaciones multiplataforma aumente. Pese a esto, se ha demostrado mediante la realización del marco de diseño de la interacción en aplicaciones móviles que es posible encontrar similitudes y equivalencias tanto en la estructura de las aplicaciones como en los elementos empleados en el diseño de la interfaz a través de diferentes plataformas. Estos factores similares entre plataformas facilitan la labor de diseñadores y desarrolladores de aplicaciones durante el proceso de desarrollo.

Por último, se ha aprendido que en el diseño software lo esencial es el usuario. Conocer cuales son los dispositivos más demandados, las aplicaciones más usadas, las características de diseño propias de cada plataforma y las similitudes entre aplicaciones de una misma plataforma es crucial para comprender cómo el usuario espera interactuar con la aplicación que se quiere desarrollar. Por todo esto si se quiere garantizar una plena experiencia de usuario debemos garantizar que el diseño de la interacción, usabilidad y accesibilidad de las aplicaciones que se desarrollan sean acordes a lo que el usuario espera en la plataforma que utiliza y para conseguir este objetivo puede ser de gran utilidad el uso del marco de diseño de la interacción creado en el presente trabajo.

9. Líneas futuras

Tras la realización del presente trabajo y como siguientes pasos en el desarrollo del marco de aplicaciones sería de gran utilidad llevar a cabo evaluaciones de otras aplicaciones móviles multiplataforma existentes basándonos en el marco, para así realizar mejoras sobre el mismo que permitan adaptarlo a las necesidades reales identificadas de su uso por profesionales del diseño de aplicaciones móviles.

Otra de las posibles tareas que se podría llevar a cabo para la mejora del mismo, sería que diseñadores y/o desarrolladores de aplicaciones móviles que están comenzando a conceptualizar un producto, tomaran como punto de partida en el diseño de la interacción el marco de diseño. Para de este modo, poder recibir realimentación de personas implicadas en el sector.

Por otra parte, la herramienta web podría ser mejorada con la información recogida tras la realización de las tareas descritas anteriormente. De forma adicional, se podrían incluir nuevas funcionalidades que permitieran, por ejemplo, la realización de planes de diseño personalizados que se pudieran guardar y usar como referencia en las diferentes etapas de proyectos.

Además, sería recomendable dar difusión al marco de diseño para de este modo aumentar el número de usuarios del mismo y poder obtener mayor información sobre su uso que permita su ampliación. Por último, sería conveniente realizar una evaluación de usabilidad con un amplio número de usuarios para poder obtener suficiente información que ayude a mejorar la herramienta.

10. Referencias

- [Angulo, 2014a] E. Angulo, X. Ferré, J. Alonso. UX & Cross-Platform Mobile Application Development Frameworks. 2014.
<http://raptor.ls.fi.upm.es/techreports/studyCPFrameworks.pdf>
- [Apple, 2014a] Apple. iOS Human Interface Guidelines, 2014.
https://developer.apple.com/library/ios/design/index.html#//apple_ref/doc/uid/TP40013289
- [Apple, 2014b] Apple. iOS 7 UI Transition Guide, 2013.
<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/TransitionGuide/TransitionGuide.pdf>
- [Barea, 2013] A. Barea, X.Ferré, L.Villaroel. Android vs. iOS Interaction Design Study for a Student Multiplatform App. HCI International 2013 - Posters' Extended Abstracts Communications in Computer and Information Science Volume 374, 2013, pp 8-12
http://link.springer.com/chapter/10.1007%2F978-3-642-39476-8_2
- [Bloq, 2013] Creative Bloq. Iconic calculator that inspired an iPhone app is re-released, 2013. <http://www.creativebloq.com/design/iconic-calculator-inspired-iphone-app-re-released-5132723>
- [Cava, 2013] Marco della Cava. Jony Ive: The man behind Apple's magic curtain, 2013. <http://www.usatoday.com/story/tech/2013/09/19/apple-jony-ive-craig-federighi/2834575/>
- [Cosmos, 2013] Cosmos. Se cumplen cinco años desde Android 1.0, 2013.
<http://www.xatakandroid.com/sistema-operativo/se-cumplen-cinco-anos-desde-android-1-0>
- [Google, 2014a] Google Inc. and the Open Handset Alliance. Android User Interface Guidelines, 2014. <http://developer.android.com/design/>
- [Google, 2014b] Google Inc. Material Design, 2014.
<http://www.google.com/design/spec/material-design/introduction.html>
- [Haveri, 2014] Brian Haveri. Responsive CSS Framework Comparison, 2014.
<http://responsive.vermilion.com/compare.php>
- [IDC, 2014] IDC. Worldwide Smartphone Shipments Edge Past 300 Million Units in the Second Quarter; Android and iOS Devices Account for 96% of the Global Market, 2014. <http://www.idc.com/getdoc.jsp?containerId=prUS25037214>
- [McNeil, 2013] Patrick McNeil. The Web Designer's Idea Book, Volume 3: Inspiration from Today's Best Web Design Trends, Themes and Styles, Volume 3, 2013.
<http://books.google.es/books?id=VVQlAgAAQBAJ&pg=PA180&lpg=PA180&dq=refer>

[rences+for+skeuomorph&source=bl&ots=6dX50WkTyl&sig=LJ8io4HT-ryIZNojXpptz2s8rv4&hl=es&sa=X&ei=3DOAVPHOOcP7ggSK5oDQAg&ved=0CEwQ6AEwBQ#v=onepage&q=references%20for%20skeuomorph&f=false](#)

[Quesada, 2013] Sarah Quesada. ¿Qué es Responsive Web Design?, 2013.
[http://www.maestrosdelweb.com/editorial/que-es-responsive-web-design/](#)

[W3C, 2012a] World Wide Web Consortium Schools. HTML5 Introduction, 2012.
[http://www.w3schools.com/html/html5_intro.asp](#)

[W3C, 2012b] World Wide Web Consortium Schools. CSS3 Introduction, 2012.
[http://www.w3schools.com/css/css3_intro.asp](#)

Este documento esta firmado por



Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Wed Jan 07 21:54:59 CET 2015
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)